MODELING AND SIMULATION OF NON-ISOTHERMAL GAS-ASSISTED INJECTION MOLDING FOR VISCOUS NEWTONIAN AND NON-NEWTONIAN FLUIDS

Undergraduate Honors Thesis

Presented in Partial Fulfillment of the Requirements for graduation "with Honors Research Distinction in Chemical and Biomolecular Engineering"

By

Panayiotis K. Kolliopoulos

*****

William G. Lowrie Department of Chemical and Biomolecular Engineering

The Ohio State University

Columbus, OH 43210

April 15th, 2016

Thesis Committee:

Dr. Kurt W. Koelling, Advisor

Dr. Isamu Kusaka

Acknowledgements

I would like to thank all of the people who have assisted me throughout the course of my research. In particular, I would like to express my extreme gratitude to my research advisors, Dr. Kurt W. Koelling and Varun Venoor. Without their guidance and support, this research experience would not have been as rewarding as it was. I would also like to thank my defense committee member Dr. Isamu Kusaka who I have worked with for a year and has greatly helped me in my academic pursuits. Finally, I would like to thank The Ohio State University Undergraduate Research Scholarship Fund for the financial assistance.

Abstract

Gas-assisted injection molding (GAIM) process is the injection of a molten polymer resin into a cooled mold cavity simultaneously or followed by the injection of pressurized gas into the resin to fill out the mold cavity and form hollow pockets in the resin. Current gas-assisted injection molding simulations do not accurately predict hydrodynamic coating thickness due to the frozen layer created by the polymer melt in non-isothermal systems. The proposed models are a hybrid control-volume finite element/finite-difference method to obtain a numerical simulation of gas-assisted injection molding for non-isothermal systems, at high capillary numbers, for Newtonian and non-Newtonian fluids. The selected viscous Newtonian fluids exhibited viscosities with high and low temperature dependencies. The non-Newtonian fluids that were selected had shear thinning viscosities that could be modeled using the power law. Previous studies employed a number of assumptions in order to simplify the simulation model, however, those assumptions contributed to a significant deviation error in predicting the coating thickness. The proposed Newtonian model was compared with experimental data and the results confirmed that the simulation was reliable for predicting maximum fractional coverage and delay time at which maximum fractional coverage occurred. On average the model has a maximum deviation of 4% in the hydrodynamic coating thickness and 15% in the predicted delay time. Additionally, the effects of tube diameter, temperature difference between the polymer melt and cooling fluid, and the heat transfer coefficient, on fractional coverage were isolated and master curves were developed over a large range of operating conditions using dimensionless variables. Due to the lack of experimental data, the non-Newtonian model was compared to previous simulations done by other researchers. Compared to existing simulation models, the recommended models are simpler and require less computing power without

compromising their accuracy, thus, functioning as an alternative approach to gas-assisted injection molding simulations. Accurately predicting hydrodynamic coating thickness has a number of applications in the plastics and polymer industry, specifically in decreasing production costs with faster cycle times, improving product quality and reducing resin costs.

Vita

2012 ……...……..…… Velo High School, Korinth, Greece

2016 ……..….. B.S. Chemical and Biomolecular Engineering, The Ohio State University

2016 to Current ……...…...…… Ph.D Chemical Engineering, University of Minnesota

Field of Study

Major Filed: Chemical and Biomolecular Engineering

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

The displacement of less viscous fluid in viscous fluid has practical applications in several commercial processes like flow through porous media in enhanced oil recovery, production of hollow fiber membranes, blood flow and activation phenomena. One important application in the polymer processing field is gas-assisted injection molding (GAIM) technology.

GAIM process is the injection of a molten resin into a mold cavity simultaneously or followed by the injection of pressurized gas into the resin to fill out the mold cavity and form a hollow pocket(s) in the resin [1] as seen in the figure below.

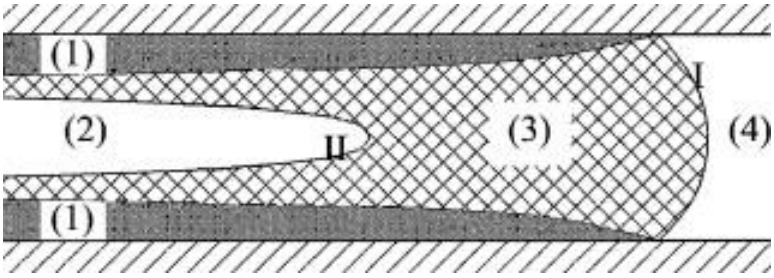

Figure 1: Schematic notation for flow regions and their interface in gas-assisted injection molding: (1) the solid frozen layer, (2) the penetration gas, (3) the deforming viscous melt, (4) the unfilled cavity, (I) the melt front, (II) the gas front [4].

The gas is then used to transit the packing pressure to compensate for polymer shrinkage and is vented out just prior to opening the mold [2-3]. GAIM makes the production of parts with complex geometry, precise dimension, high surface quality and high strength to weight ratio possible. Additional advantages include reduction of weight and cycle times, smaller holding pressure and less clamp forces, which results in substantial material and process savings. Also the process provides higher flexibility in part design, especially for complex parts with thick and thin sections.

With the gas penetration through a viscous fluid, the displacement causes instability of the interface between the two fluids, resulting in a single long bubble penetrating into the viscous

fluids. The coating layer thickness of the viscous fluid left on the inner wall depends on the bubble velocity, flow geometry and the flow field property, for which temperature and fluid rheological properties are the critical factors. Our study will concentrate on how these factors will affect the coating thickness.

Due to the fact that mold design and process control are so critical and difficult, computer simulation of GAIM is necessary. The GAIM simulation software that is available is MOLDFLOW, MOLDEX3D and others. All of these software packages are based on a midplane model or a 3D model. The disadvantage of the midplane model is that a second modelling is inevitable. The 3D model requires a full-scale three-dimensional discretization of thin parts which results in unsustainable computing time and instability of calculation [4].

The purpose of this study is to develop an alternative simulation models for predicting the coating thickness, for viscous Newtonian and non-Newtonian fluids under non-isothermal systems in capillary tubes for a range of high capillary numbers.

Chapter 2: Literature Review

Section 2.1: Newtonian Experiments

There are a number of theoretical and experimental studies that have been conducted on the GAIM process. Initial research was conducted by Fairbrother and Stubbs [5] on Newtonian systems. They determined that the fractional coverage remaining in the system is related to the ratio of the viscous and surface forces in the system. This ratio is known as the Capillary Number (*Ca*) and is defined in equation 1 below, where η is the viscosity of the fluid, *U* is the velocity of the fluid and σ is the surface tension.

$$Ca = \frac{\eta U}{\sigma} \tag{1}$$

Taylor [6] carried out experiments to determine the relationship between coating thickness and Capillary Number and extended the work done by Fairbrother and Stubbs up to Capillary number values of 2.0. Cox [7] improved on Taylor's work and determined that the asymptotic value of the fractional coverage at high capillary numbers was around 0.6. A theoretical solution was first provided by Bretherton [8] to predict the process of bubble penetration through Newtonian fluid at low capillary numbers. Reinelt [9] and Kolb and Cerro [10] obtained numerical solutions for Newtonian fluids in close agreement with data. Due to these advances in research, GAIM, became industrially important, however, most of the fluids used are non-Newtonian.

Section 2.2: Non-Newtonian Experiments

Initial research using non-Newtonian fluids was conducted by Poslinski [11]. Poslinski carried out experiments of gas bubble penetrating through Non-Newtonian fluid in capillary tube, under isothermal conditions. An attempt to numerically simulate the penetration of a constant velocity gas bubble through an isothermal, shear thinning fluid was conducted later by Poslinski

and Coyle [12]. Huzyak and Koelling [13] studied isothermal gas penetration of elastic (Boger) fluids that exhibited Newtonian shear viscosity in circular tubes using experiments. Gauri and Koelling [14] extended that study of long bubbles penetrating through viscoelastic material in capillary tubes. Polynkin [15] thoroughly experimented on gas penetration in tubes. Chen [16] improved computer times for existing simulation methods. Shen [17] studied gas penetration in a 2D gas channel system. Zhou [18] extended Bretherton's model for high capillary numbers. Belblidia [19] developed correlations for predicting the coating thickness in a 2D straight cylindrical tube. Li [4] developed a surface model for the simulation of filling process in GAIM.

In this paper an alternative simulation model for predicting the coating thickness is suggested, for viscous Newtonian and non-Newtonian fluids under non-isothermal systems in capillary tubes for a range of high capillary numbers. The advantages of the proposed model include the model simplicity and the short process time, compared to existing models. The model is based on using the Finite Element Method (FEM) and Finite Difference Method (FDM) to manipulate governing and derived equations with the purpose of determining fractional coverage. The Newtonian model results were compared with experimental data from Tendulkar [20] and the non-Newtonian model results were compared with simulation results from Poslinski and Coyle [12] for verification.

Chapter 3: Experimental Apparatus

Gas-assisted injection molding, as a polymer processing technique, is usually conducted under non-isothermal conditions. To simplify the process, Tendulkar [18] used a capillary tube. During the bubble penetration, when the outer temperature was low enough, a frozen layer will form on the inner wall during the penetration. The setup provided results in order to calculate the fractional coverage in the experiment of a bubble penetrating through non-isothermal viscous fluid which behaved as a Newtonian fluid isothermally. The apparatus used by Tendulkar [20] was an extension of the apparatus used by Huzyak and Koelling [13], the experiments were carried out using precision bore stainless steel tubes, each with a length of 30.5 cm. Three tubes with outer diameters of 1.27 cm, 0.9525 cm, and 0.635 cm were used in the experiments. The corresponding internal diameters were 1.0922 cm, 0.7747 cm, and 0.4928 cm respectively. The inlet of each tube was connected through a three-way valve and flexible stainless steel hoses to a gas reservoir and a pressure vessel filled with the polymer. There was a small stainless steel flow restriction tube connected to the tube via Swagelok fittings to control the volume flow rate of polymer during the gas penetration at an approximate constant. The diameter and length of the restriction tube was selected so that the ratio of the final volume flow rate to the original volume flow rate was less than 1.1.

Two water baths were maintained at different constant temperatures. The tube filled with polymer was first immersed into an initial water bath; after the polymer inside the tube achieved uniform temperature distribution, the tube was taken out and put into the other water bath. It was assumed that when the Fourier number, *Fo*, becomes equal to 1, the temperature distribution will be homogenous. The Fourier number was defined as:

$$Fo = \frac{\alpha t}{R_{in}^2} \tag{2}$$

Where α is the thermal diffusivity, t is the time and R_{in} is the inner radius of the tube.

After the tube stayed in the second water bath for a certain delay time, the three-way valve was switched to the gas reservoir and a gas bubble penetrated through the Newtonian fluid, pushing part of the fluid out of the tube. The measured parameter was the hydrodynamic fractional coverage, *m*, which was defined as the fraction of the cross sectional area of the tube coated with fluid:

$$m = 1 - \frac{R_b^2}{R_{in}^2} \tag{3}$$

Where R_b is the radius of the gas bubble, R_{in} is the inner radius of the tube. The weight of the fluid that had been pushed out was measured to calculate the fractional coverage. Assuming a constant fractional coverage along the length of the tube, equation 2 can be expressed as:

$$m = 1 - \frac{w}{\pi R_{in}^2 L \rho_p} \tag{4}$$

Where w is the weight of fluid that has been displaced, L is the length of the tube and ρ_p is the test fluid density.

The experiments were done at delay times ranging from 0 to 500 sec, which corresponded to Fourier number $0<Fo<1$. The experiments were carried out at very high bubble penetration rate, which can make the change of temperature gradient during the penetration negligible, so it was assumed that the temperature only varied in the radial direction and thus made the heat transfer model in one dimension. The capillary number range was from 20 to 1000. There were five temperature gradients formed between the two water baths, from the one in which the tube was first immersed in, to the one where the gas-assisted injection experiments were carried out. They were 25°C-0°C, 50°C-0°C, 35°C-25°C, 50°C-25°C and 65°C-25°C.

Chapter 4: Fluid Characterization

Two fluids were used in this study. The test fluids were a high molecular weight polybutene (Amoco Polybutene H-300, $M_w \approx 1340$ g mol^{-1}) and silicone oil (Dow Corning, DC 200). The shear viscosity was measured using the Rheometrics Fluid Spectrometer, RFSII. Viscosity was measured using a 25 mm couette tool with a temperature controlled bath in the steady shear sweep test. The tests were carried out under temperature ranging from 10°C to 60°C. The data showed the fluids were Newtonian and temperature sensitive, as can be seen in figures 2 and 3 below.

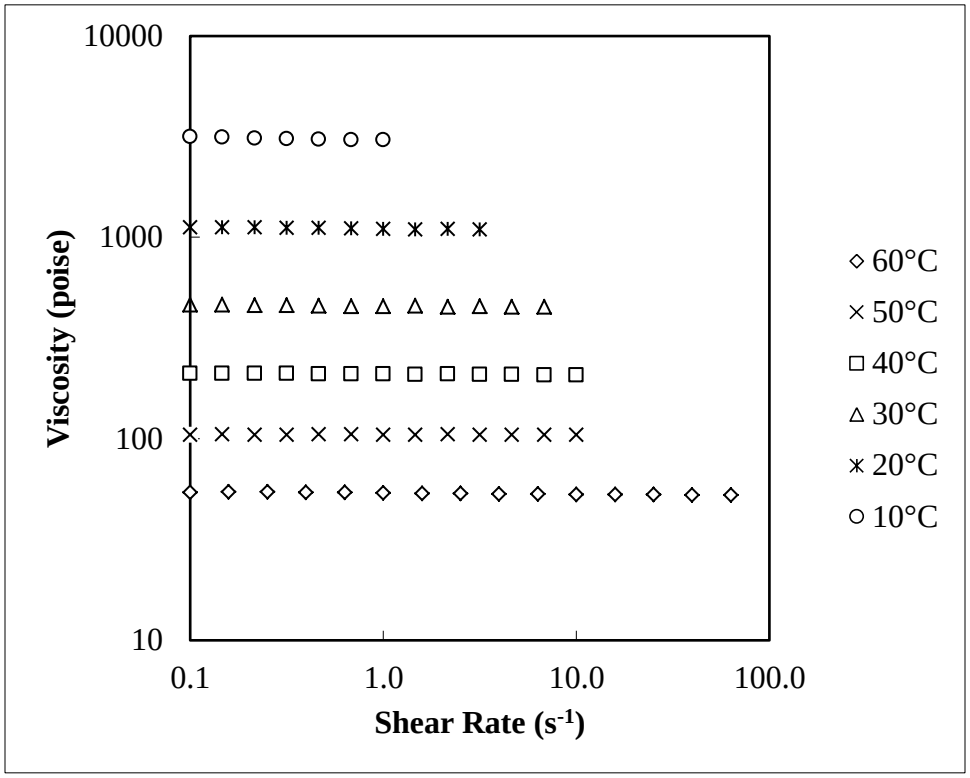

Figure 2: Shear behavior of PBH-300 for temperatures ranging from 10^0C to 60^0C.

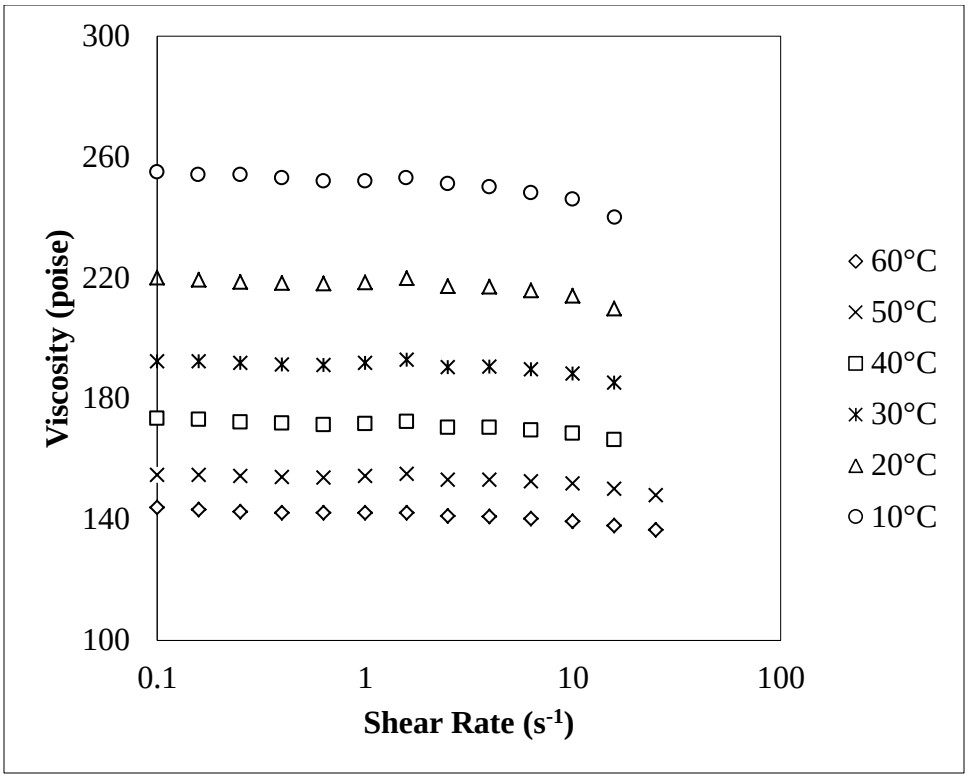

Figure 3: Shear behavior of DC-200 for temperatures ranging from 10^0C to 60^0C.

The temperature sensitivity of each fluid can be seen in figures 2 and 3 above. It was observed that PBH-300 is extremely temperature sensitive, whereas, DC-200 has small temperature sensitivity.

The viscosity of the polymer was determined using the Arrhenius model defined as:

$$\eta = Aexp\left(\frac{\Delta H}{RT}\right) \tag{5}$$

Where η is the shear viscosity, A is the frequency factor, $\Delta H/R$ is the flow activation energy and T is the temperature. The density of PBH-300 has been reported by the Amoco Chemical Company [21], and the variation of density with temperature was reported to be:

$$\rho_p = -0.58T + 903 \tag{6}$$

Where ρ_p is the polymer density and T is the temperature of the polymer in degrees Celsius. The physical properties of the two test fluids are seen in table 1, below.

Table 1: Physical properties of PBH-300 and DC-200, at a reference temperature of 25°C.

Properties	Symbol	PBH-300	DC-200
Density (kg/m^3)	ρ_p	888.5	971
Thermal Conductivity (W/m K)	k_p	0.1125	0.1547
Specific Heat (J/kg K)	$C_{p,p}$	2100	1456
Surface Tension (N/m)	σ	0.0272	0.0215
Flow Activation Energy (K)	$\Delta H/R$	8575	1090
Zero Shear Viscosity (Pa-s)	η_0	72.2	20.5
Frequency Factor (Pa-s)	A	2.33×10^{-11}	0.5296

Chapter 5: Modeling of Newtonian fluids

Although the gas-assisted injection molding process has many advantages as a polymer processing technique, it is a very complicated process. The objective of this section is to develop a model to predict the fractional coverage of Newtonian fluids which depends on the temperature gradient, delay time in gas injection, tube diameter and flow activation energy. Based on previous simulation done, when the capillary number is greater than 10, the fractional coverage will achieve a constant asymptotic value of 0.6.

To theoretically predict the change of fractional coverage with delay time, a frozen layer model was employed. In figure 4, below, the isothermal and non-isothermal gas injection process are compared.

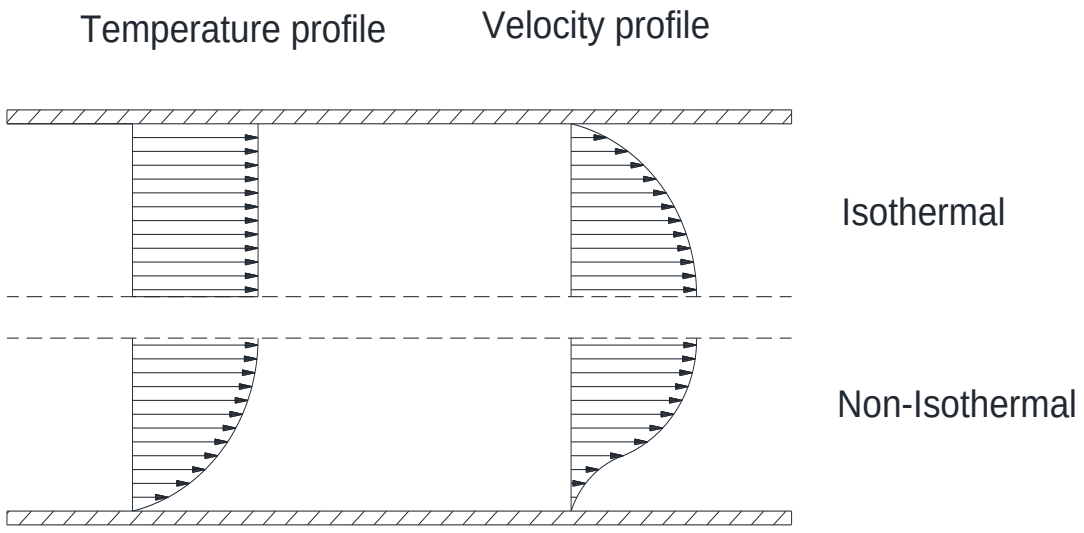

Figure 4: Comparison of Isothermal and non-Isothermal temperature and velocity profiles for Newtonian fluids.

As it is observed, the continuous change of temperature in radial direction caused the velocity profile to deviate from parabolic shape. This was the origin of the difference between the results observed in the isothermal and non-isothermal experiments. In the frozen layer model,

it was assumed that there was always a frozen layer formed during the bubble penetration through the non-isothermal polymer melt. The fluid inside the frozen layer has a uniform temperature, which is the inner temperature of the polymer melt obtained during the stay of the tube filled with polymer in the first water bath. Thus the fluid inside the frozen layer can be treated as a homogenous Newtonian fluid. All the experiments were carried out under high capillary numbers in order to achieve the asymptotic value of 0.6.

Finite difference method was employed to calculate the thickness of the frozen layer. The system was divided into three heat transfer regions. The regions are shown in figure 5. Each region was divided into 21 elements, which was visually determined to be the minimum number of nodes needed to accurately capture the temperature gradients.

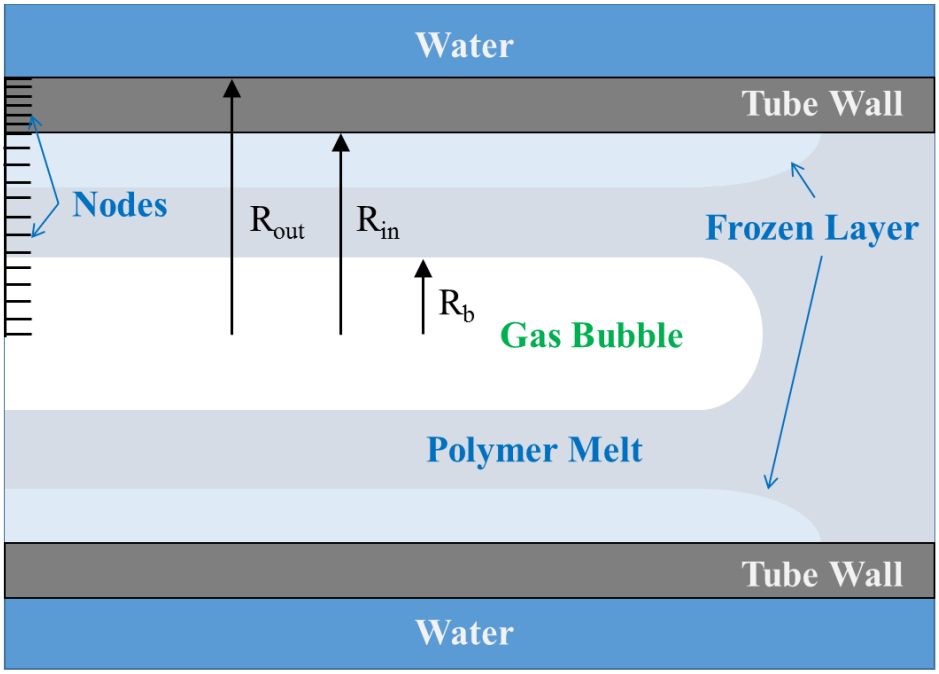

Figure 5: One dimensional heat transfer model with the heat transfer region divided into three parts.

Section 5.1: Heat transfer outside the tube

The heat transfer outside the tube was a continuous process across the whole water bath region. This situation can be simplified by assuming the heat transfer just occurred in a region very near the tube outer wall. In other words, it was assumed that there was a thin layer attached to the tube wall and the heat transfer resistance concentrates in it. Thus the governing equation for this region is:

$$-k_s \left.\frac{\partial T_s}{\partial r}\right|_{r=R_{out}} = h(T_w - T_B) \tag{7}$$

Where k_s is the thermal conductivity of stainless steel, R_{out} is the outer tube radius, h is the convective heat transfer coefficient of the assumed thin layer coating the tube, T_w is the wall temperature and T_B is the temperature of the bulk fluid.

Using the finite element method, equation 8 becomes:

$$T_{s(N,j+1)} = T_{s(N,j)} + \frac{2a_s dt}{dr_2^2} * \left[T_{s(N-1,j)} - T_{s(N,j)} - \left(\frac{R_{out} + \frac{dr_2}{2}}{R_{out}} \right) \left(\frac{h dr_2}{k_s} \right) (T_{s(N,j)} - T_B) \right] \tag{8}$$

Where $T_{s(N,j)}$ is the temperature of the steel/water interface at time step j, dt is the specified time step, dr_2 is the radial step change in the steel which is defined in the appendix, a_s is the thermal diffusivity of the steel and k_s is the thermal conductivity of the steel.

Due to the fact that the rate of water flowing in and out of the water bath was very small compared to the size of the water bath, the effect of forced convection heat transfer was neglected and the main heat transfer type considered was free convection. The convective heat transfer coefficient, h, was determined by using the following equation by Churchill and Chu [22].

$$Nu = \left[0.60 + 0.387\left(\frac{Ra}{[1 + (0.559/Pr)^{9/16}]^{16/9}}\right)^{1/6}\right]^2 \quad (9)$$

Where *Nu* is the Nusselt number, *Ra* is the Rayleigh number and *Pr* is the Prandtl number, which are defined in the appendix.

Section 5.2: Heat transfer inside stainless steel

The governing equations for this region are:

$$\frac{\partial T_s}{\partial t} = \alpha_s\left(\frac{\partial^2 T_s}{\partial r^2} + \frac{1}{r}\frac{\partial T_s}{\partial r}\right) \quad (10)$$

Where T_s is the temperature of the steel tube and α_s is the thermal diffusivity of the steel.

Using the finite element method, equation 10 becomes:

$$T_{s(i,j+1)} = T_{s(i,j)} + \frac{a_s dt}{r_{s(i)} dr_2^2} * \left[\left(\frac{r_{s(i)} + r_{s(i-1)}}{2}\right)T_{s(i-1,j)} - 2r_{s(i)}T_{s(i,j)} + \left(\frac{r_{s(i+1)} + r_{s(i)}}{2}\right)T_{s(i+1,j)}\right] \quad (11)$$

Where $T_{s(i,j)}$ is the temperature of the steel at node *i* at time step *j*, $r_{s(i)}$ is the radial position in the steel at node *i*.

The boundary conditions employed in this region were:

BC1: $$T_{s(i,0)} = T_i$$

BC2: $$T_{s(N,j)} = T_w$$

BC3:
$$T_{s(i,j+1)} = T_{s(i,j)} + \frac{2k_s dt\left(R_{in} + \frac{dr_2}{2}\right)\left(T_{s(i+1,j)} - T_{s(i,j)}\right)}{dr_2\left[\left(R_{in} + \frac{dr_2}{4}\right)\rho_s C_{p,s} dr_2 + \left(R_{in} - \frac{dr_1}{4}\right)\rho_p C_{p,p} dr_1\right]} + \frac{2k_p dt\left(R_{in} - \frac{dr_1}{2}\right)\left(T_{s(N-1,j)} - T_{s(i,j)}\right)}{dr_1\left[\left(R_{in} + \frac{dr_2}{4}\right)\rho_s C_{p,s} dr_2 + \left(R_{in} - \frac{dr_1}{4}\right)\rho_p C_{p,p} dr_1\right]}$$

Section 5.3: Heat transfer inside polymer

The governing equations for this region are:

$$\frac{\partial T_p}{\partial t} = \alpha_p \left(\frac{\partial^2 T_p}{\partial r^2} + \frac{1}{r} \frac{\partial T_p}{\partial r} \right) \tag{12}$$

Where T_p is the temperature of the polymer and α_p is the thermal diffusivity of the polymer.

Using the finite element method, equation 12 becomes:

$$T_{p(i,j+1)} = T_{p(i,j)} + \frac{a_p dt}{r_{p(i)} dr_1^2} * \left[\left(\frac{r_{p(i)} + r_{p(i-1)}}{2} \right) T_{p(i-1,j)} - 2 r_{p(i)} T_{p(i,j)} + \left(\frac{r_{p(i+1)} + r_{p(i)}}{2} \right) T_{p(i+1,j)} \right] \tag{13}$$

Where $r_{p(i)}$ is the radial position in the polymer at node *i* and dr_1 is the radial step change in the polymer.

The boundary conditions employed in this region were:

BC4: $$T_{p(i,0)} = T_i$$

BC5: $$T_{p(i,j+1)} = T_{p(i,j)} + 4 \left(\frac{a_p dt}{dr_1^2} \right) \left(T_{p(i+1,j)} - T_{p(i,j)} \right)$$

The temperature profile far in front of the penetrating bubble was calculated, and to improve the model, the temperature profiles of polymer near the bubble were taken into consideration. It was assumed that as the bubble penetrates forward at a high velocity, the fluid in front of the bubble was directly squeezed into the region between the bubble and inner wall. Also, the transition region between the bubble tip and static flow field far in front of the bubble was neglected for simplicity of calculation by assuming the temperature profile at the tip of the bubble was the same as that at the far end of the tube. The region between the bubble tip and edge was divided into small segments. It was assumed that the shape of the bubble penetrating through isothermal Newtonian fluid at very high capillary number could be also used for the calculation under non-isothermal conditions. Thus by finding the radial position of the bubble curve at each node, the squeezed temperature profile could be calculated. Then by averaging the

temperature values of each profile, a new temperature profile was determined that corresponded to that between the bubble tip and edge.

Section 5.4: Velocity profile of polymer

In the calculation of the velocity profile, pseudo steady state was assumed. The equation of motion was set up for a non-isothermal system, thus incorporating the effect of temperature on viscosity, which was done using equation 5. The governing equation used to solve for the velocity profile was:

$$\frac{\partial}{\partial r}\left(r\eta\frac{\partial u_z}{\partial r}\right)=\frac{\Delta Pr}{L} \tag{14}$$

Where ΔP is the pressure gradient and L is the tube length.

Using the finite difference method, equation 14 becomes:

$$u_{z(i,j)}=u_{z(i-1,j)}+\frac{1}{2(N-1)}\left(\frac{\Delta Pr_{p(i)}}{2LA}\right)\left[\frac{r_{p(i)}}{\exp\left(\frac{\Delta H}{RT_{p(i,j)}}\right)}+\frac{r_{p(i-1)}}{\exp\left(\frac{\Delta H}{RT_{p(i-1,j)}}\right)}\right] \tag{15}$$

The boundary conditions used were:

BC6: $$u_z|_{r=R_{in}}=0$$

BC7: $$\left.\frac{du_z}{dr}\right|_{r=0}=0$$

Using the temperature profile between the bubble tip and edge the velocity profile could be calculated, using equation 15.

Section 5.5: Hydrodynamic fractional coverage

In order to calculate the fractional coverage, the velocity profile had to be normalized. A new parabolic shaped velocity profile was assumed using the equation depicted below:

$$u^* = 1 - \left(\frac{r}{R_x}\right)^2 \tag{16}$$

Where u^* is the normalized velocity profile of the "equivalent" parabolic shaped profile and R_x is the new radius that could be related to the thickness of the coating layer and is shown in figure 6 below.

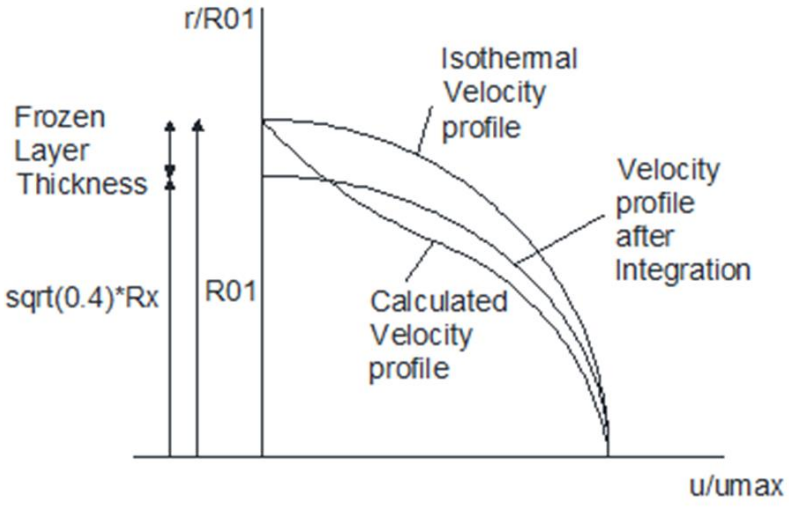

Figure 6: Isothermal, calculated and assumed velocity profiles.

The equivalence of volume flow rate was assumed to relate the calculated velocity profile to the new parabolic shaped profile, as is depicted in the following equation.

$$2\pi \int_0^{R_{in}} r u_z^* dr = 2\pi \int_0^{R_x} r u^* dr \tag{17}$$

Where u_z^* is the calculated normalized velocity. Thus by using equations 15 and 16, the new radius, R_x could be determined. Once the new radius was determined the hydrodynamic fractional coverage could be calculated using the following equation:

$$m = 1 - \left(\frac{\sqrt{1 - m_{\infty}} R_x}{R_{in}} \right)^2 \tag{18}$$

Where m_{∞} is the asymptotic value of the fractional coverage, which based on experimental data, is 0.6 at high Capillary numbers (Ca>10), as seen in figure 7 below.

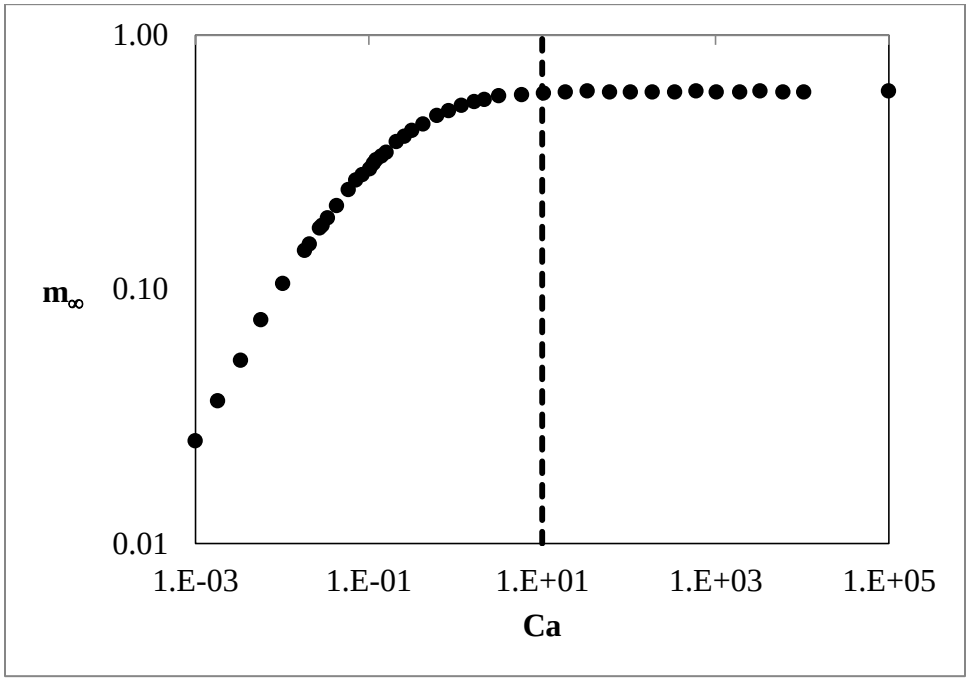

Figure 7: Fractional coverage vs. Capillary number for Newtonian fluids, by Poslinski and Coyle [12].

This method was implemented in Code 1 and 2 which can be found in the appendix and whose results are discussed in the next chapter.

Chapter 6: Simulation Results for Newtonian fluids under Non-Isothermal Conditions

The simulation results in this section are based on the Newtonian fluid model presented in Chapter 5. The simulations results were compared with the experimental results obtained by Tendulkar [20]. The test fluids used for the experiments and simulations were polybutene H-300 (PBH-300) and silicon oil DC-200. The simulation was done using MATLAB.

Simulations were conducted for all the experimental conditions used by Tendulkar. The experimental trials can be seen in the table below.

Table 2: Experimental Trials conducted by Tendulkar [20].

OD (cm)	ID (cm)	$T_i - T_B$				
		25°C-0°C	35°C-25°C	50°C-0°C	50°C-25°C	65°C-25°C
0.635	0.2464	PBH-300	-	-	PBH-300	-
0.9525	0.3874	-	-	PBH-300	-	-
1.27	0.5461	PBH-300	PBH-300	PBH-300	PBH-300	PBH-300 DC-200

L (cm)	ΔP (Pa)	k_s (W/(m*K))	ρ_s (kg/m^3)	$C_{p,s}$ (J/(kg*K))
30.5	1.013×10^6	16	7850	514.5

Due to the fact that all the simulation results had good agreement with the experimental data and in order to prevent repeatability, a select few of the simulation results will be mentioned in this chapter and the rest can be found in the appendix.

In addition to comparing the simulation results to experimental data, this study tried to isolate the effects of parameters such as tube diameter, initial polymer and cooling fluid temperatures, flow activation energy and Biot number (*Bi*).

Section 6.1: Effect of tube diameter on fractional coverage

In order to determine the effects of the tube diameter on fractional coverage the simulation results using OD (1.27 cm and 0.635 cm) and T_i-T_B (50°C-25°C) were compared to experimental data and are shown in figure 8 below.

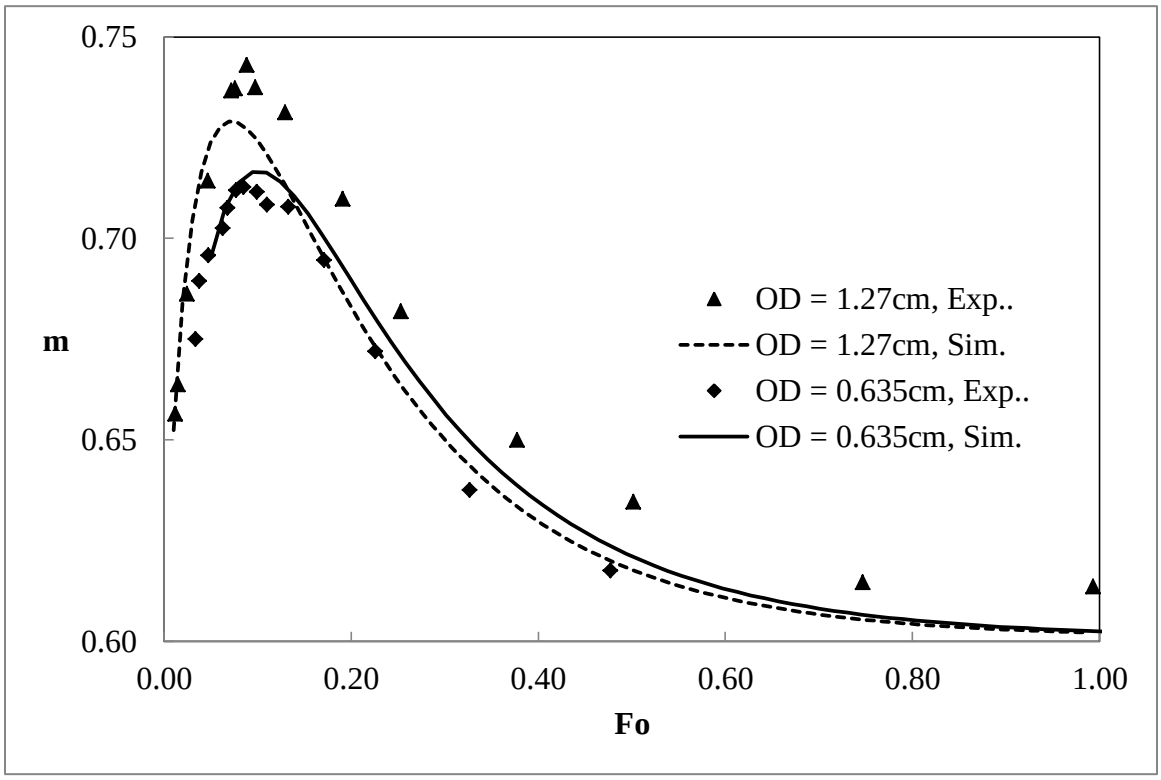

Figure 8: Simulation and Experimental results of Fractional Coverage vs. Fourier Number for OD (1.27 cm and 0.635 cm) and T_i-T_B (50°C-25°C).

As seen in the figure above, the model accurately predicts the maximum fractional coverage as well as the time at which the maximum fractional coverage occurs. The maximum deviation of the maximum fractional coverage between the model and the experimental data for 1.27 cm and 0.635 cm is 2% and 1% respectively. Additionally, the maximum deviation between Fourier number at which the maximum fractional coverage occurs between the model and the experimental data for 1.27 cm and 0.635 cm is 21.2% and 11.4% respectively. The

simulation model and experimental results for 1.27 cm and 0.635 cm at 25°C-0°C and for 1.27 cm and 0.9525 cm at 65°C-25°C can be found in the appendix.

Based on the simulation and experimental results it was determined that increasing the tube diameter results in an increase in the maximum fractional coverage. However, the increase in the tube diameter did not affect the Fourier number at which maximum fractional coverage occurs due to the fact that the Fourier number includes the diameter and therefore, the diameter effects are normalized. Nevertheless, as the tube diameter was increased, there was an increase in the time at which maximum fractional coverage occurred.

Once the effect of the diameter had been determined, the study continued by trying to isolate the effects of the initial polymer and cooling fluid temperatures on fractional coverage.

Section 6.2: Effect of initial polymer and cooling fluid temperatures on fractional coverage

In order to determine the effects of the initial polymer and cooling fluid temperatures the simulation results using OD (1.27 cm) and T_i-T_B (35°C-25°C, 50°C-0°C and 50°C-25°C) were compared to experimental data and are shown in figure 9 below.

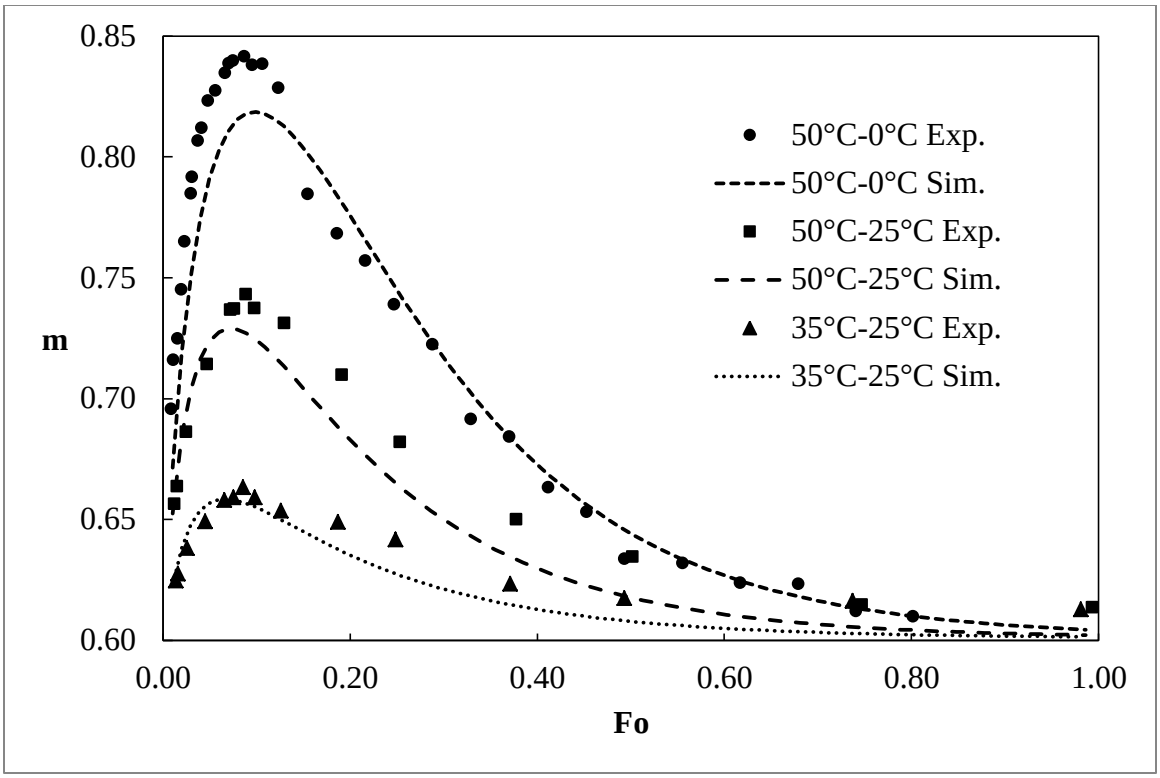

Figure 9: Simulation and Experimental results of Fractional Coverage vs. Fourier Number for OD (1.27 cm) and T_i-T_B (35°C-25°C, 50°C-0°C and 50°C-25°C).

As it is observed from the figure above, the model accurately predicts the maximum fractional coverage as well as Fourier number at which the maximum fractional coverage occurs. The maximum deviation of the maximum fractional coverage between the model and the experimental data for 35°C-25°C, 50°C-0°C and 50°C-25°C is 1%, 3% and 2% respectively. Additionally, the maximum deviation between Fourier number at which the maximum fractional coverage occurs between the model and the experimental data for 35°C-25°C, 50°C-0°C and 50°C-25°C is 20%, 14.6% and 21.2% respectively. The simulation model and experimental results comparison for 25°C-0°C and 65°C-25°C using OD of 1.27 cm can be found in the appendix.

Based on the simulation and experimental results it was determined that increasing the temperature gradient (T_i-T_B) results in an increase in the maximum fractional coverage. However,

it was challenging to attribute the change of the maximum fractional coverage to the increase of the initial polymer temperature (T_i) or to the decrease of the cooling fluid temperature (T_B).

Therefore, once having a general understanding of how the initial temperature gradient affected the fractional coverage, the study continued with evaluating the effect of the flow activation energy on fractional coverage.

Section 6.3: Effect of flow activation energy on fractional coverage

In order to determine the effects of the flow activation energy the simulation results using OD (1.27 cm) and T_i-T_B (65°C-25°C) for PBH-300 and DC-200 were compared to experimental data and are shown in figure 10 below.

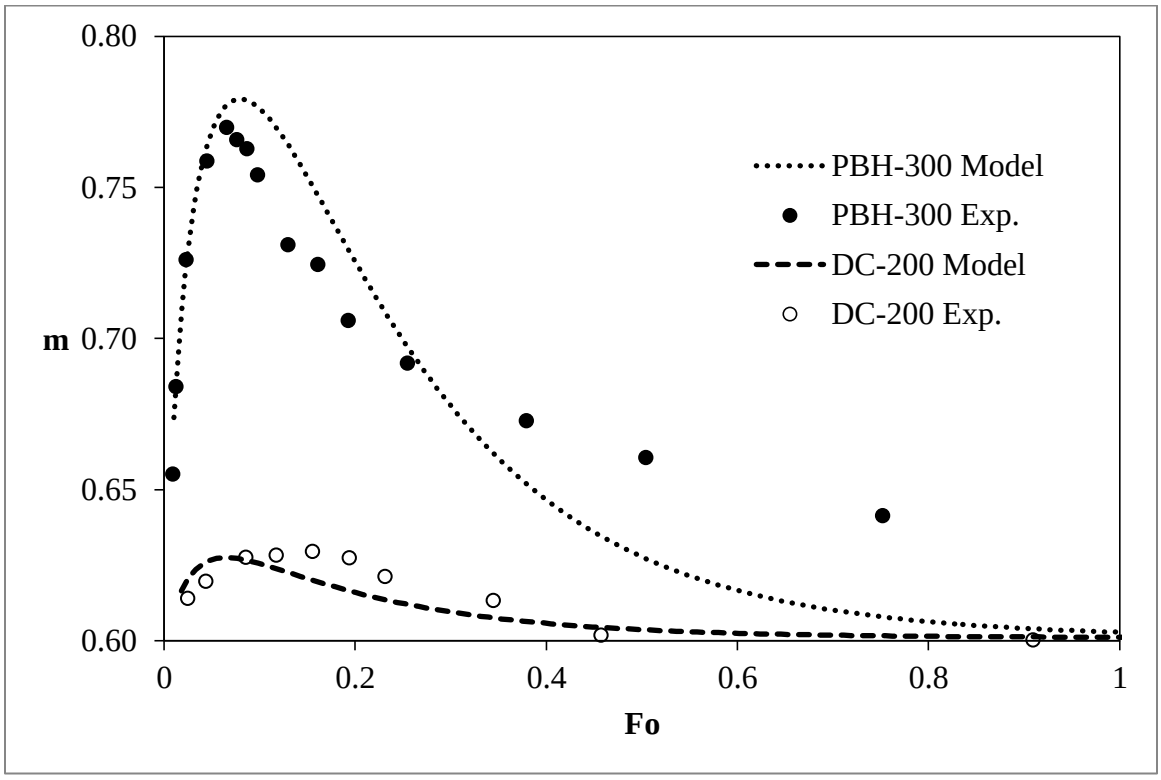

Figure 10: Simulation and Experimental results of Fractional Coverage vs. Fourier Number for OD (1.27 cm) and T_i-T_B (65°C-25°C) for PBH-300 and DC-200.

Similarly to the other simulations, the model accurately captures the maximum fractional coverage. However, the focus of this section is the effect of the flow activation energy on fractional coverage. The flow activation energy of PBH-300 and DC-200 is 8575 K and 1090 K respectively. It is clear that the flow activation energy has a large impact on the magnitude of the maximum fractional coverage. Therefore, because PBH-300 is extremely temperature sensitive compared to DC-200, the magnitude of the maximum fractional coverage for PBH-300 is much greater than that of DC-200.

Section 6.4: Effect of Biot number on fractional coverage

The Biot number, (*Bi*) is a dimensionless variable that gives an index of the ratio of the heat transfer resistances inside and at the surface of the body and is defined as:

$$Bi = \frac{hR_{in}}{k_p} \tag{19}$$

A number of simulations were conducted to determine the effect of the Biot number on fractional coverage and the results are depicted in the figure below.

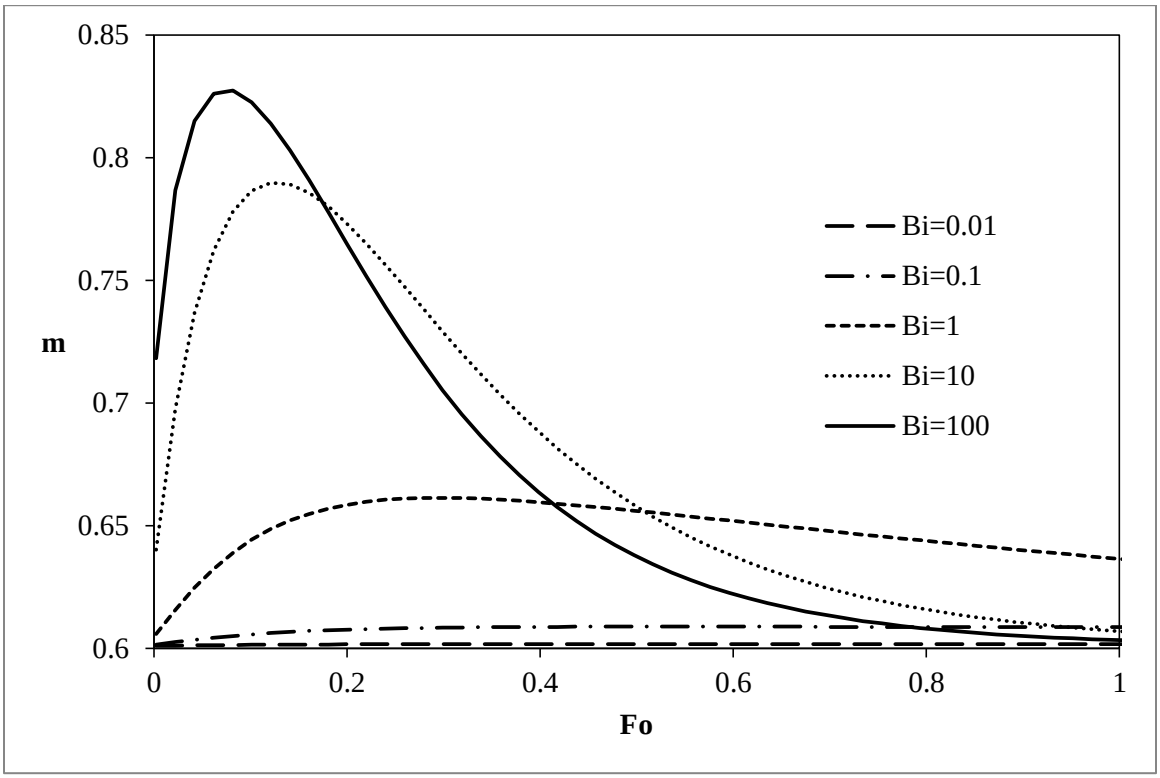

Figure 11: Simulation results of Fractional Coverage vs. Fourier Number for OD (1.27 cm) and T_i-T_B (50°C-0°C) for a range of Biot numbers.

Based on the simulation results, it is observed that as the Biot number increases, then the fractional coverage increases. The reason is that the larger the Biot number, the larger resistance there is to heat transfer inside the polymer and therefore, there is a larger temperature gradient across the polymer, resulting in a larger fractional coverage. Almost all the previous simulations that have been shown in this study are at a Biot number between 10 and 100.

Section 6.4: Isolating effect of temperature difference on fractional coverage

The study continued with the purpose of creating a master curve, where one could extract data using a specified T_i-T_B, $\Delta H/R$, *Bi*, and *OD* and could predict the fractional coverage at any Fourier number. Therefore, in order to create this master curve, the data for a specified T_i-T_B, $\Delta H/R$, *Bi*, and *OD* would have to collapse, which as is seen from the figure below, the selection of T_i and T_B matters, because even though T_i-T_B is the same for all three curves, the fractional coverage is different.

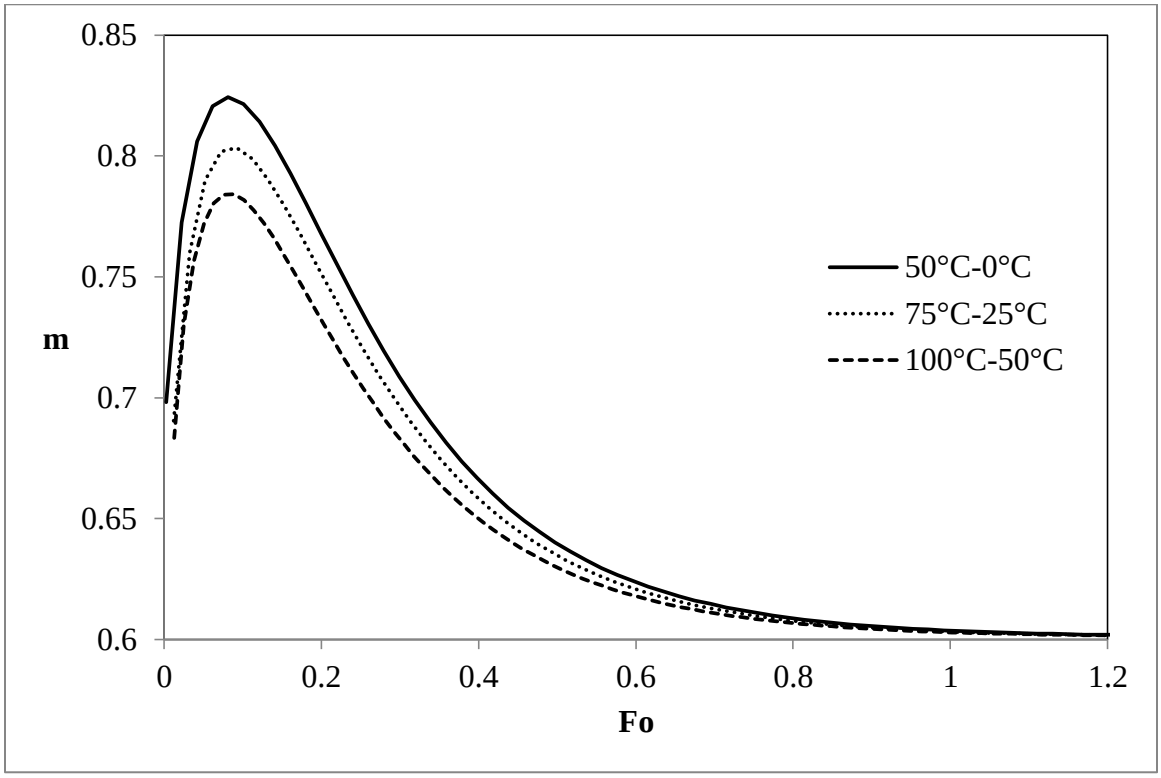

Figure 12: Simulation results of Fractional Coverage vs. Fourier Number for OD (1.27 cm) and Biot number (50) for T_i-T_B (50°C-0°C, 75°C-25°C, and 100°C-50°C).

So a large number of simulations were conducted with the purpose of determining the dependence of the T_i-T_B regardless of the individual T_i and T_B values. So for a Biot number of 50 the following curve was created.

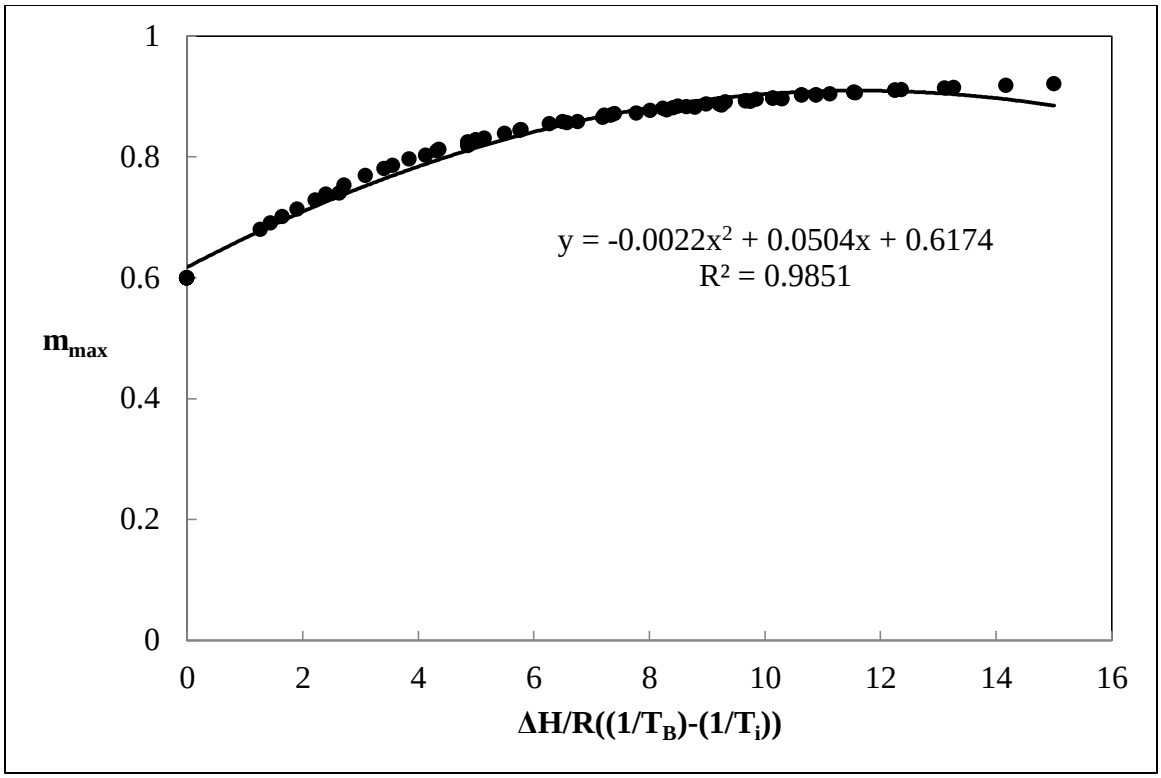

Figure 13: Maximum fractional coverage vs. $\Delta H/R((1/T_B)-(1/T_i))$ for Biot number of 50.

Therefore, by using this curve for a Biot number of 50, the maximum fractional coverage can be determined without performing the simulation itself. It is important to note that, this curve depends on the Biot number that is selected, and therefore, if experiments were done at a different Biot number, a new curve would have to be created. Increasing the Biot number increases the dependence of the maximum fractional coverage on temperature gradient selection, which can be seen in the figure below.

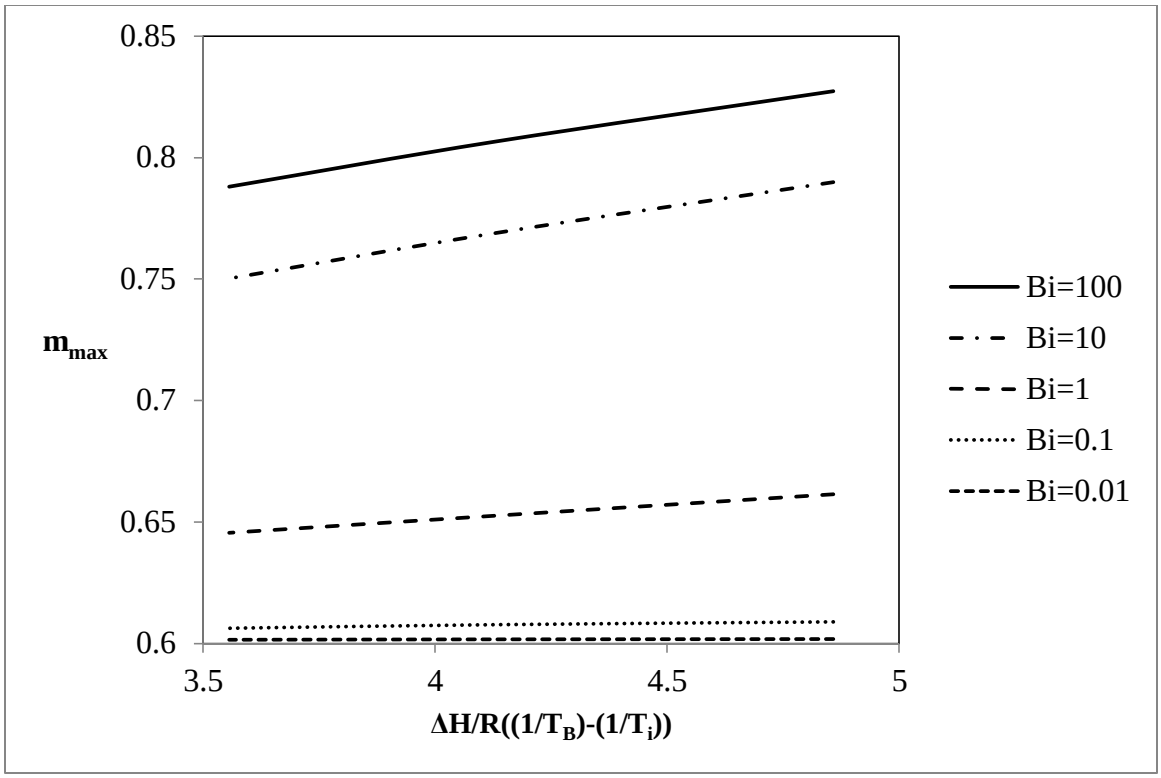

Figure 14: Maximum fractional coverage vs. $\Delta H/R((1/T_B)-(1/T_i))$ for a range of Biot numbers.

The next step was to normalize the fractional coverage, in order to make the data collapse. This was done by introducing the variable, normalized fractional coverage, m^*, where:

$$m^* = \frac{m - m_\infty}{m_{max} - m_\infty} \tag{20}$$

By changing the y-axis in figure 12 from fractional coverage to normalized fractional coverage, the following figure is created.

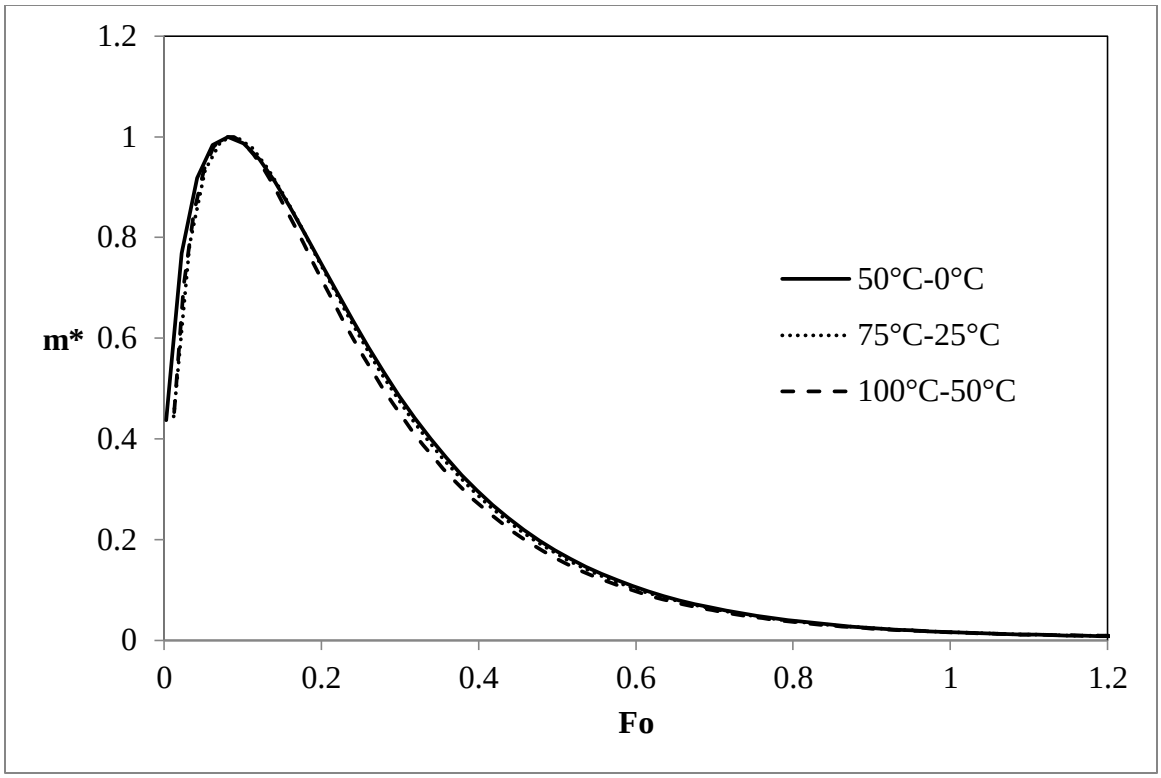

Figure 15: Normalized fractional Coverage vs. Fourier Number for OD (1.27 cm) and Biot number (50) for T_i-T_B (50°C-0°C, 75°C-25°C, and 100°C-50°C).

Therefore, it is observed that the data collapses and by specifying T_i-T_B, $\Delta H/R$, *Bi*, and *OD*, one could predict the fractional coverage regardless of the individual values of T_i and T_B.

With a solid fundamental understanding of how each parameter affected the fractional coverage, then the model was expanded for non-Newtonian shear thinning fluids whose viscosity follows the power law.

Chapter 7: Modeling of Non-Newtonian fluids

The model that was developed for non-Newtonian fluids, and specifically Power Law fluids, is a slight variation of the model for Newtonian fluids. The heat transfer calculations were the same, however, the viscosity now varies with shear rate, $\dot{\gamma}$, along with temperature. The viscosity for Power Law fluids is defined as:

$$\eta = K\dot{\gamma}^{n-1} \tag{21}$$

Where:

$$K = A\exp\left(\frac{\Delta H}{RT}\right) \tag{22}$$

And

$$\dot{\gamma} = \frac{\partial u_z}{\partial r} \tag{23}$$

Therefore, by using equations 15, 21-23 and the finite difference method, the velocity is calculated using:

$$u_{z(i,j)} = u_{z(i-1,j)} + \frac{1}{2(N-1)}\left(\frac{\Delta P r_{p(i)}}{2LA}\right)^{\frac{1}{n}}\left[\left(\frac{r_{p(i)}}{\exp\left(\frac{\Delta H}{RT_{p(i,j)}}\right)}\right)^{\frac{1}{n}} + \left(\frac{r_{p(i-1)}}{\exp\left(\frac{\Delta H}{RT_{p(i-1,j)}}\right)}\right)^{\frac{1}{n}}\right] \tag{24}$$

Where n is the power law index of the fluid. The boundary conditions used in this model were the same as those used for the Newtonian fluid model.

Once the velocity profile is calculated, similarly to the Newtonian model, a new parabolic shaped profile is assumed and using the equivalence of volume flow rate in equation 17, the new radius, R_x, was determined. Therefore, the fractional coverage can now be determined. Two different methods were done in order to select the asymptotic value of the fractional coverage. The first method suggests that the asymptotic value is 0.6, as if the fluid if Newtonian. The second method is to assume that the fluid is no longer Newtonian, and the asymptotic value of the fractional coverage is no longer 0.6. For this reason, simulations by Poslinski and Coyle [12]

were used in order to obtain the relationship between the power law index and the asymptotic value of fractional coverage for high capillary numbers. The relationship that was determined is seen in figure 15 below.

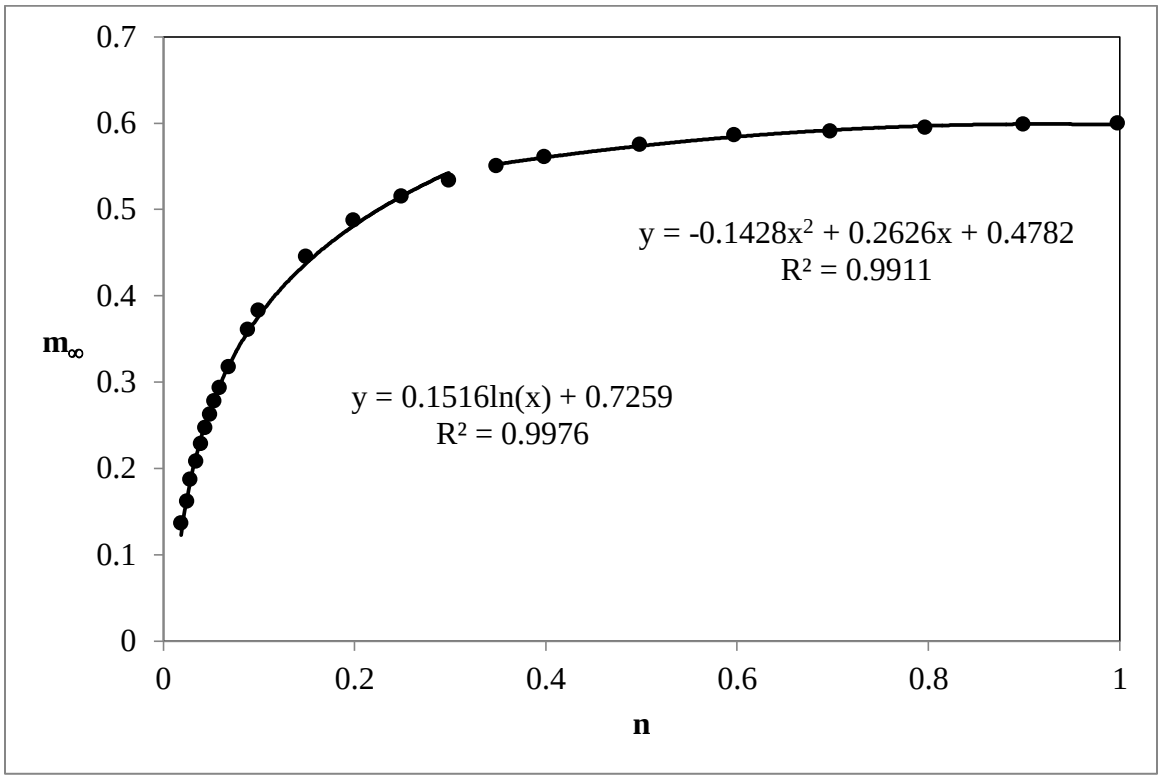

Figure 16: Fractional coverage vs. power law index for Capillary number of 1000, obtained from Poslinski and Coyle [12].

Therefore, the following relationship was obtained:

$$m_\infty = \begin{cases} 0.1516\ln(n) + 0.7259, & n \le 0.3 \\ -0.1428n^2 + 0.2626n + 0.4782, & n > 0.3 \end{cases} \quad (25)$$

Additionally, the equation 18 needed to be modified to account for the power law index in order to account for these new asymptotic fractional coverage values. The fractional coverage equation that was used in the model was:

$$m = 1 - \left(\frac{\sqrt{1 - m'_\infty}R_x}{R_{in}}\right)^2 \quad (26)$$

Where:

$$m'_{\infty} = \frac{m_{\infty} - 1 + \left(\frac{2+2n}{1+3n}\right)}{\left(\frac{2+2n}{1+3n}\right)} \tag{27}$$

Equation 27 was derived using the power law viscosity and the constant volumetric flow rate assumption.

This method was implemented in Code 3 which can be found in the appendix and whose results are discussed in the next chapter.

Chapter 8: Simulation Results for Non-Newtonian fluids under Isothermal and Non-Isothermal Conditions

The simulation results in this section are based on the non-Newtonian fluid model presented in Chapter 7. This section includes simulations done under isothermal and non-isothermal conditions. However, due to the lack of experimental data, the isothermal simulations were compared to those done by Poslinski and Coyle [12] and the non-isothermal simulations were done qualitatively.

Since in Chapter 6 the effects of parameters such as tube diameter, initial polymer and cooling fluid temperature and flow activation energy, on fractional coverage were determined, the study focused initially on the isothermal system in order to comprehend the effect of the power law index on fractional coverage.

Section 8.1: Effect of power law index on fractional coverage

The effect of the power law index was isolated by running the simulations under isothermal conditions and comparing those results to those obtained by Poslinski and Coyle. The results of those simulations, depicting the relationship between the asymptotic fractional coverage values versus the power law index can be seen in figure 16 below.

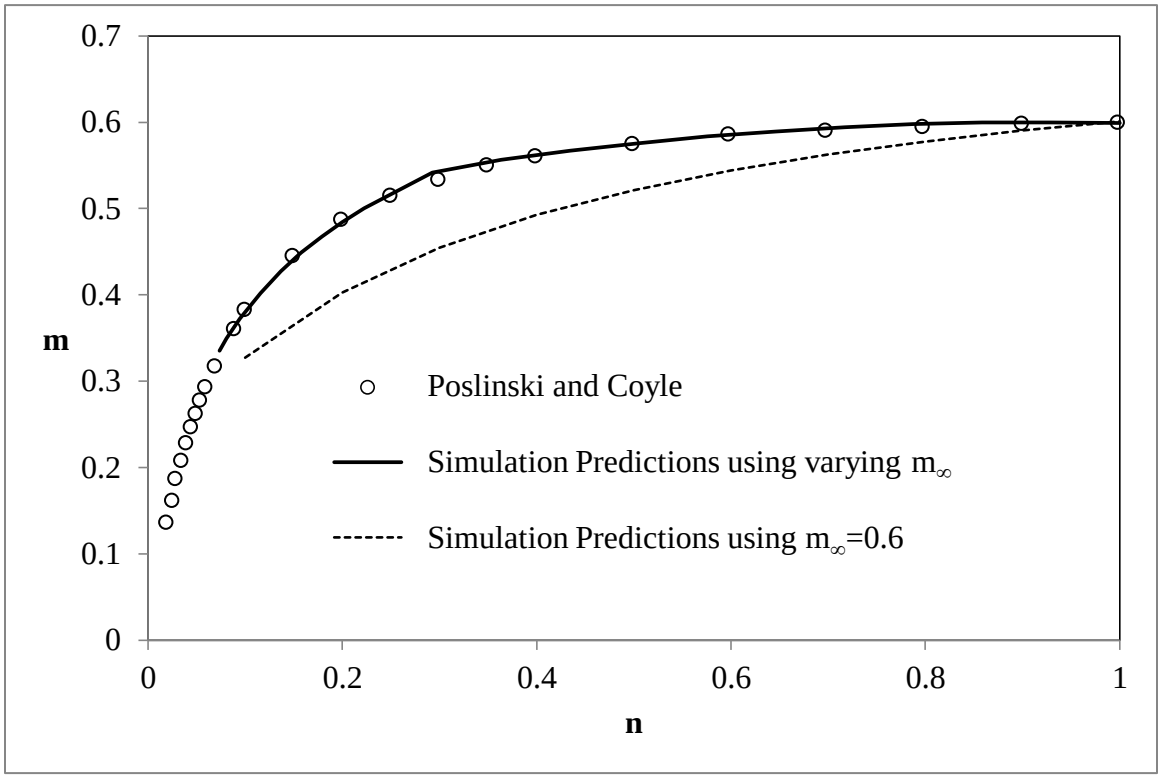

Figure 17: Fractional coverage vs. power law index of data from Poslinski and Coyle [12], simulation predictions using varying m_∞ and simulation predictions using m_∞=0.6.

The simulation predictions that used the Newtonian asymptotic fractional coverage value of 0.6 are compared with the simulation predictions that used varying asymptotic fractional coverage values based on data from Poslinski and Coyle and with simulation results from Poslinski and Coyle. Based on the figure above, the model using m_∞=0.6 does a reasonable prediction however, the model using m_∞ by Poslinski and Coyle, matches the simulation results that Poslinski and Coyle simulated quite accurately. However, the model is limited to a certain

power law index value (around 0.05) due to the fact that the inverse of the power law index is included in the exponent in the model and the smaller the power law index becomes, the larger the number MATLAB needs to compute. Therefore, the simulation returns an error value because of lack of memory storage. Nevertheless, the simulation results showed that the proposed method works.

Section 8.2: Effect of non-isothermal conditions on fractional coverage

The study continued to qualitatively predict the fractional coverage for non-Newtonian fluids under non-isothermal conditions. The model that was selected was the one using the varying asymptotic fractional coverage values based on data from Poslinski and Coyle. The results are shown in figure 17 below.

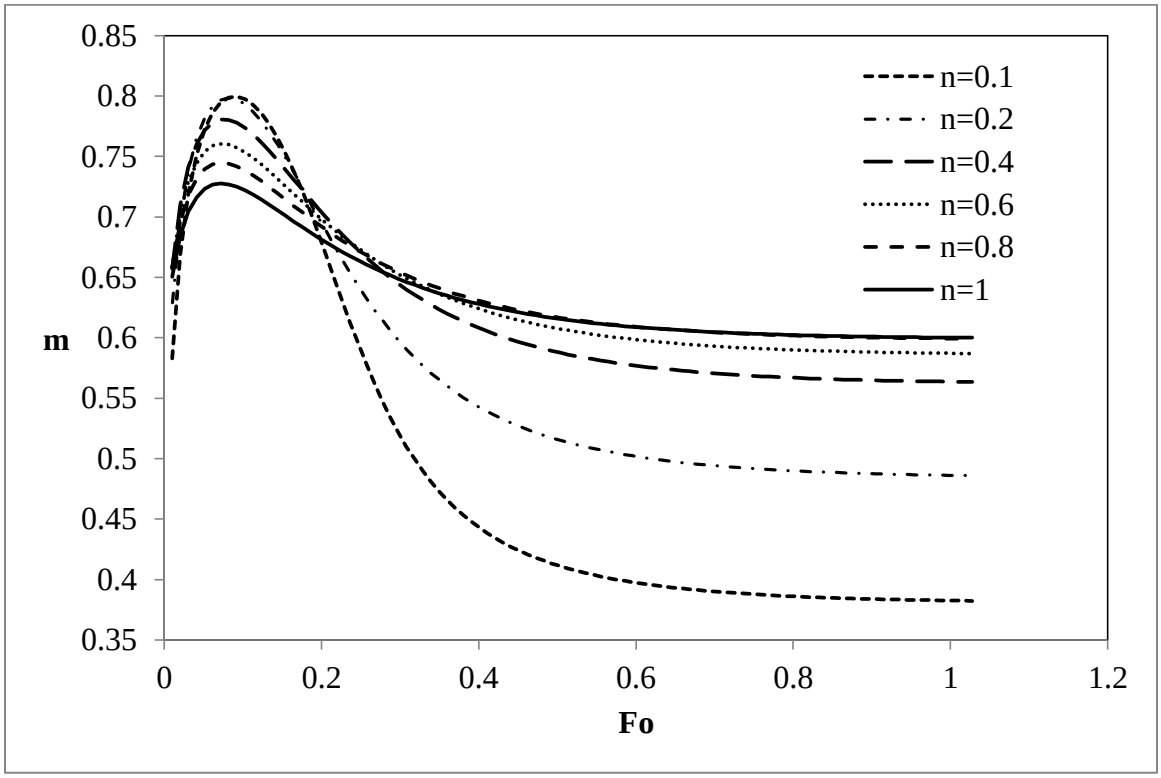

Figure 18: Simulation results of Fractional Coverage vs. Fourier Number for OD (1.27 cm) and T_i-T_B (50°C-25°C) for a range of Power Law fluid indexes.

Qualitatively it is observed that the maximum fractional coverage value decreases as the power law index increases, which is due to the fact that the fluid is become more shear thinning. However, as it was observed in figure 16 and 17 the asymptotic value of the fractional coverage increases are the power law index increases. Therefore, the simulation results are qualitatively accurate; however, in order to make a quantitative verification of the results, experimental data is required.

Chapter 9: Conclusions

Two simulation models for predicting fractional coverage using gas-assisted injection of in capillary tubes were presented. The first model was for Newtonian fluids, and the second was for non-Newtonian fluids whose viscosity could be fitted to the power law. The Newtonian simulation model was compared with experimental results performed by Tendulkar [20]. The comparison with the experimental results showed that the Newtonian simulation model was reliable and the process time was very short. In addition to comparing the simulation results to experimental data, the effects of parameters such as tube diameter, initial polymer and cooling fluid temperature difference, flow activation energy and Biot number were isolated and studied. It was determined that an increase of either the tube diameter, initial polymer and cooling fluid temperature difference, flow activation energy or Biot number results in an increase in the fractional coverage. Additionally, it was determined that at a specified tube diameter, Biot number, flow activation energy and initial polymer and cooling fluid temperature difference, a normalized fractional coverage can be calculated and is not affected by the individual values of the initial polymer and cooling fluid temperatures.

The non-Newtonian model, on the other hand was evaluated qualitatively, due to the lack of experimental data. Nevertheless, two methods with different asymptotic fractional coverage values were compared with simulation results by Poslinski and Coyle [12]. Once a more accurate method was selected, simulations for non-isothermal systems were conducted. It was determined that as the power law index increases, the maximum fractional coverage decreases, but the asymptotic fractional coverage increases.

Chapter 10: Future Work

It must be noted that the models presented have four limitations. First, the models are based on the assumption that the hydrodynamic fractional coverage, at high capillary numbers, is uniform along the length of the tube. If the capillary numbers were not as high, then the models would not be as effective. Second, the models are limited to capillary tubes and therefore, would not work for an alternative geometry. Third, the models are limited to Newtonian fluids and non-Newtonian power law shear thinning fluids. Finally, the models cannot predict the occurrence of gas blow out, which is when the gas bubble is too fast and instead of displacing the polymer, it goes through it and surpasses it, which results in defective molded parts.

The model can be expanded to eliminate these limitations. In the case that the uniform fractional coverage assumption is not made, then the bubble and polymer interaction must be taken into account and modeled, in order to determine the path of least resistance that the bubble will follow. A similar method must be used in the case that alternative geometries want to be studied.

A similar approach can be used to expand the model for non-Newtonian power law shear thickening fluids or for non-Newtonian fluids whose viscosity can be modeled using the Cross or Ellis model. The Ellis model method would be recommended as the calculations are comparable to those of the power law model, whereas the Cross model calculations are more complex.

References

[1] Loren, N. S. (1993). *U.S. Patent No. 5,204,050*. Washington, DC: U.S. Patent and Trademark Office.

[2] Isayev, A. I., Shell, R. I., & Hall, E. L. (2000). *Molding Processes in Handbook of Industrial Automation*.

[3] Avery, J. (2001). *Gas-assist injection molding: principles and applications*. Hanser Verlag.

[4] Li, J., Chen, L., Zhou, H., & Li, D. (2009). Surface Model Based Modeling and Simulation of Filling Process in Gas-Assisted Injection Molding. *Journal of Manufacturing Science and Engineering*, **131**(1), 011008.

[5] Fairbrother, F., & Stubbs, A. E. (1935). 119. Studies in electro-endosmosis. Part VI. The "bubble-tube" method of measurement. *Journal of the Chemical Society* (Resumed), 527-529.

[6] Taylor, G. I. (1961). Deposition of a viscous fluid on the wall of a tube. *Journal of Fluid Mechanics*, **10**, 161-165

[7] Cox, B. G. (1962). On driving a viscous fluid out of a tube. *Journal of Fluid Mechanics*, **14**(01), 81-96.

[8] Bretherton, F. P. (1961). The motion of long bubbles in tubes. *Journal of Fluid Mechanics*, **10**(02), 166-188.

[9] Reinelt, D. A. (1987). The rate at which a long bubble rises in a vertical tube. *Journal of Fluid Mechanics*, **175**, 557-565.

[10] Camp, C. E., Kolb, W. B., Sublette, K. L., & Cerro, R. L. (1990). The measurement of square channel velocity profiles using a microcomputer-based image analysis system. *Experiments in Fluids*, **10**(2-3), 87-92.

[11] Poslinski, A. J., Oehler, P. R., & Stokes, V. K. (1995). Isothermal gas-assisted displacement of viscoplastic liquids in tubes. *Polymer Engineering & Science*, **35**(11), 877-892.

[12] Poslinski, A. J., & Coyle, D. J. (1994). Steady gas penetration through non-Newtonian liquids in tube and slit geometries: isothermal shear thinning effects. *Proc. Polymer Processing Society, 10th Annual Meeting* (p. 219).

[13] Huzyak, P. C., & Koelling, K. W. (1997). The penetration of a long bubble through a viscoelastic fluid in a tube. *Journal of Non-Newtonian Fluid Mechanics*, **71**(1), 73-88.

[14] Gauri, V., & Koelling, K. W. (1999). The motion of long bubbles through viscoelastic fluids in capillary tubes. *Rheologica acta*, **38**(5), 458-470.

[15] Polynkin, A., Pittman, J. F. T., & Sienz, J. (2005). Gas assisted injection molding of a handle: Three-dimensional simulation and experimental verification. *Polymer Engineering & Science*, **45**(8), 1049-1058.

[16] Chen, S. C., Hu, S. Y., Chao, S. M., & Chien, R. D. (2000). Simulation of a mold-cooling process for gas-assisted injection molded parts designed with a top rib on the gas channel. *Polymer Engineering & Science*, **40**(3), 595-606.

[17] Shen, Y. (2001). The study on polymer melt front, gas front and solid layer in filling stage of gas-assisted injection molding. *International communications in heat and mass transfer*, **28**(1), 139-148.

[18] Zhou, H., & Li, D. (2003). Filling simulation and gas penetration modeling for gas-assisted injection molding. *Applied Mathematical Modelling*, **27**(11), 849-860.

[19] Belblidia, F., Pittman, J. F. T., Polynkin, A., & Sienz, J. (2005). Gas displacing viscous shear thinning liquids from tubes: Effects of cooling before gas injection. *Chemical engineering science*, **60**(17), 4953-4956.

[20] Tendulkar, M. R. (1997). *Gas bubble penetration through newtonian fluids under non-isothermal conditions*.

[21] Amoco Chemical Company, Amoco Polybutene, Bulletin PB12-N D0394, 32 (1994).

[22] Churchill, S. W., & Chu, H. H. (1975). Correlating equations for laminar and turbulent free convection from a horizontal cylinder. *International Journal of Heat and Mass Transfer*, **18**(9), 1049-1053.

[23] Wang, Y. (2003). *The Effect of Non-Newtonian Rheology on Gas-Assisted Injection Modling Process.* The Ohio State University: PhD. Thesis.

Appendices

Nomenclature

Fo,	Fourier number;
t,	delay time [s];
R,	radius of tube [m];
m,	hydrodynamic fractional coverage;
w,	weight [kg];
L,	tube length [m];
A,	frequency factor [kg/m . s];
$\Delta H/R$,	flow activation energy [°K];
T,	temperature [°C];
k,	thermal conductivity [W/m . °K];
h,	convective heat transfer coefficient [W/m^2 . °K];
Nu,	Nussert number, hD/k;
Ra,	Rayleigh number, $PrGr$;
Pr,	Prandtl number, ν/α;
Gr,	Grashof number, $g\beta(T_s - T_b)D^3/\nu\alpha$;
g,	acceleration due to gravity [m/s^2];
D,	diameter of tube [m];
C_p,	specific heat [J/kg . °K];
u,	velocity [m/s];
u^*,	normalized velocity, u/u_{max}
Bi,	Biot number, hR_{in}/k_p;

Greek symbols

α,	thermal diffusivity [m^2/s];
ρ,	density [kg/m^3];
η,	shear viscosity [kg/m . s];
β,	coefficient for thermal expansion [°K^{-1}];
ΔT,	temperature difference [°C];
ν,	kinematic viscosity [m^2/s], η/ρ;
ΔP,	pressure difference [Pa];

Subscripts

in,	inner wall of tube;
b,	bubble;
s,	solid;

p,	polymer;
out,	outer wall of tube;
w,	wall;
B,	bulk;
i,	initial;
x,	new and assumed;
∞,	value as $t \rightarrow \infty$;
max,	maximum;
0,	value as *shear rate*$\rightarrow$*0*;
z,	coordinate system direction;

Figures

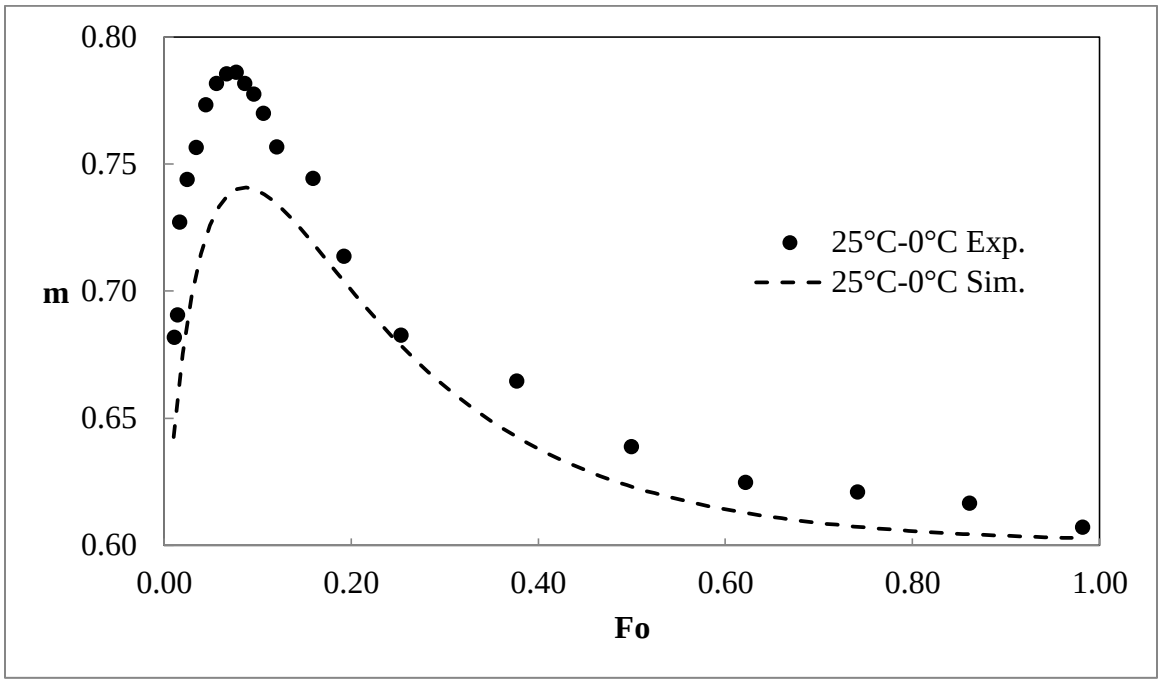

Figure 19: Simulation and Experimental results of Fractional Coverage vs. Fourier Number for OD (1.27 cm) and T_i-T_B (25°C-0°C).

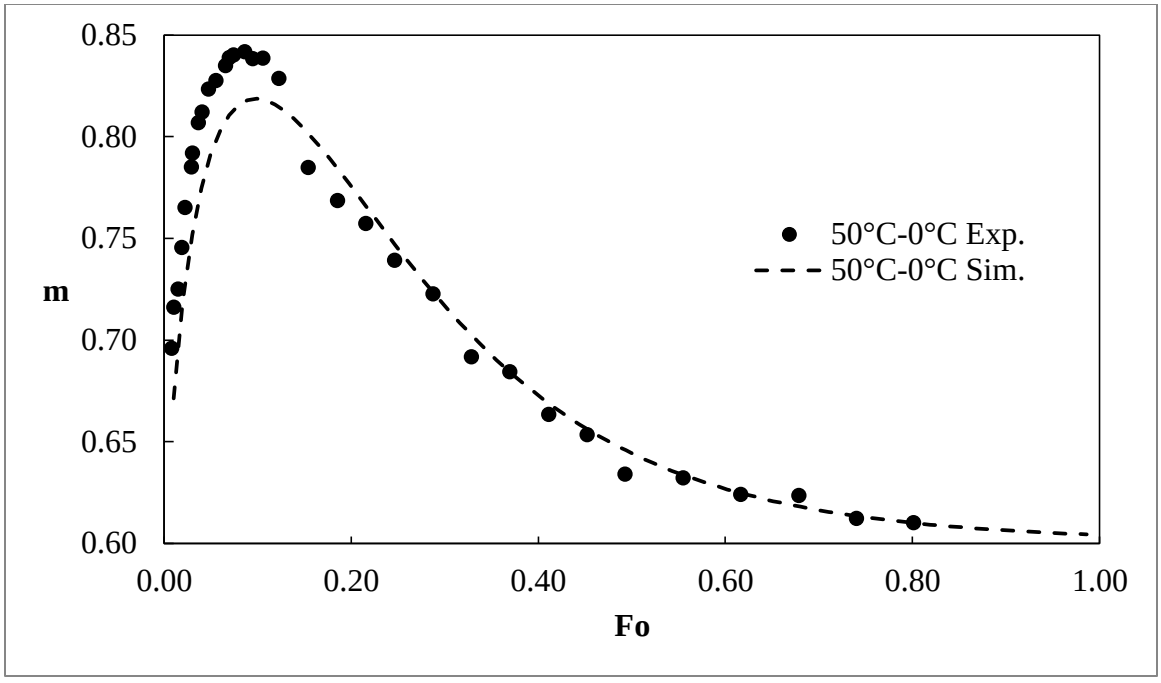

Figure 20: Simulation and Experimental results of Fractional Coverage vs. Fourier Number for OD (1.27 cm) and T_i-T_B (50°C-0°C).

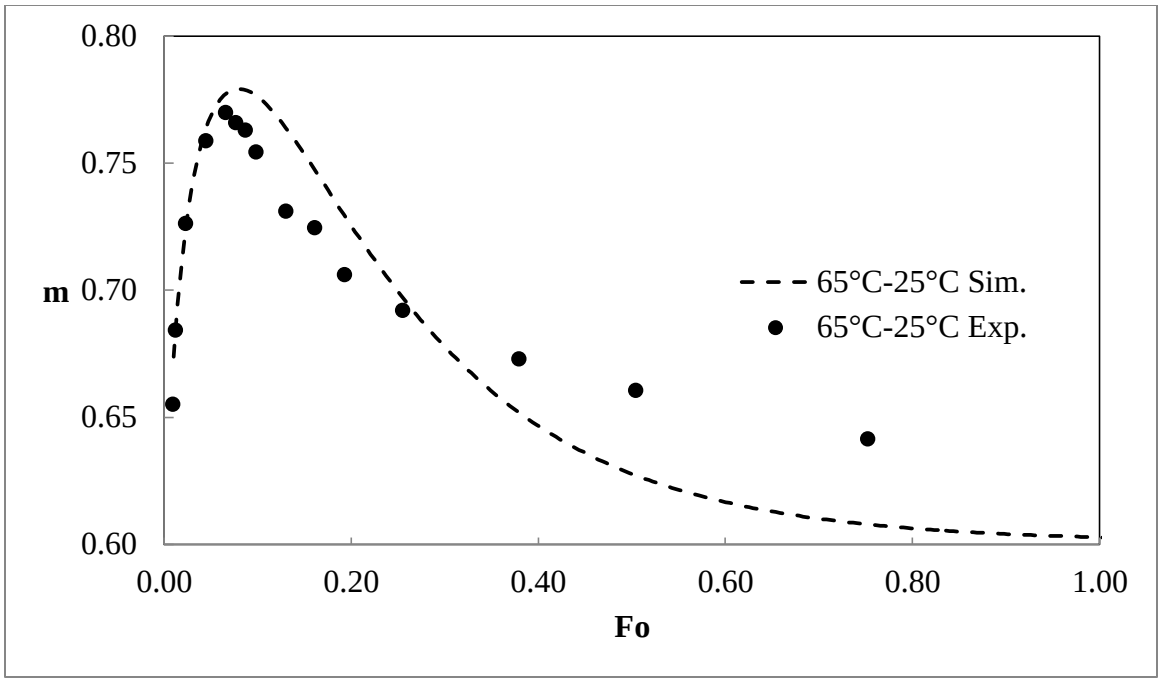

Figure 21: Simulation and Experimental results of Fractional Coverage vs. Fourier Number for OD (1.27 cm) and T_i-T_B (65°C-25°C).

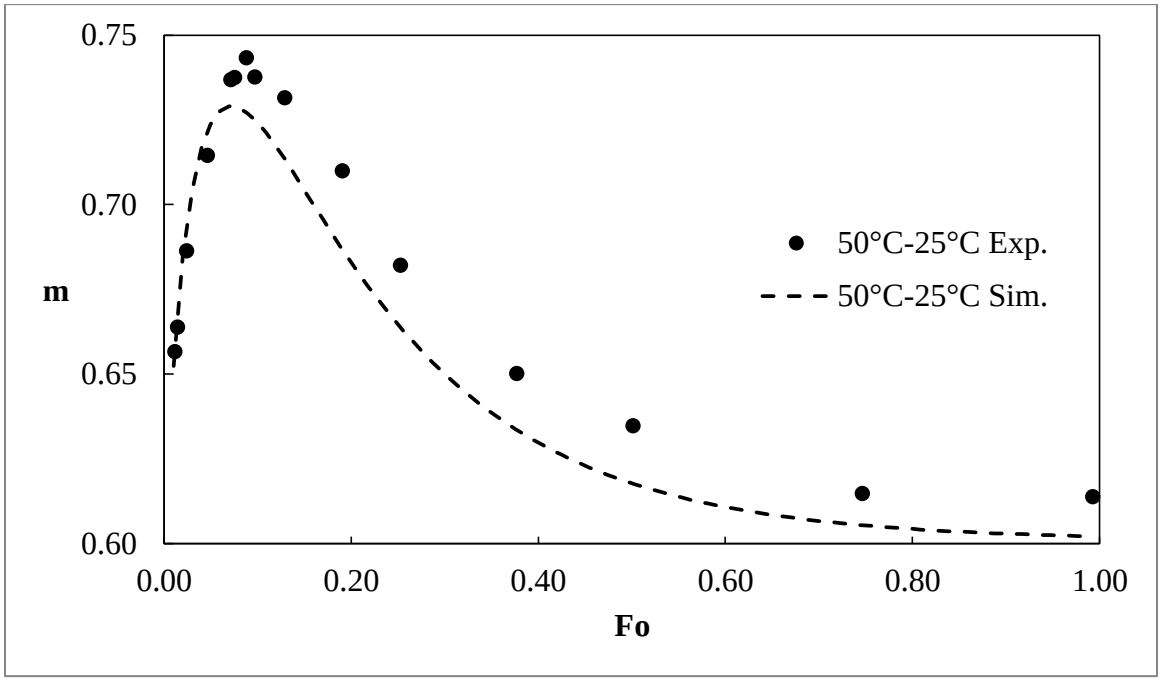

Figure 22: Simulation and Experimental results of Fractional Coverage vs. Fourier Number for OD (1.27 cm) and T_i-T_B (50°C-25°C).

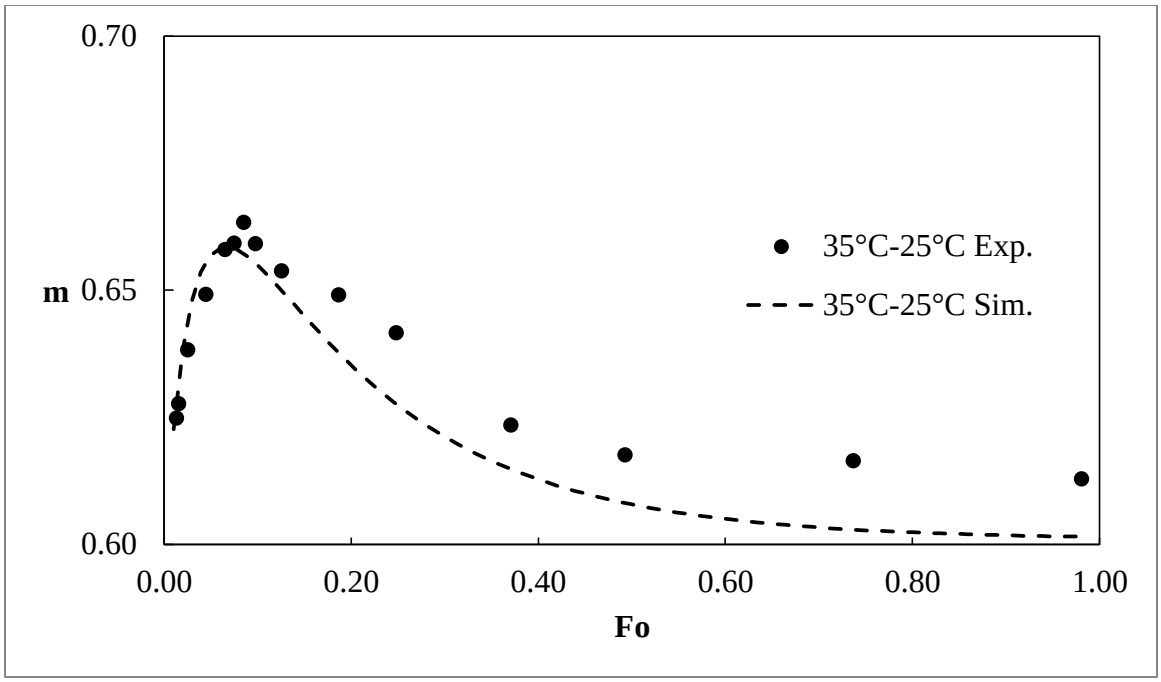

Figure 23: Simulation and Experimental results of Fractional Coverage vs. Fourier Number for OD (1.27 cm) and T_i-T_B (35°C-25°C).

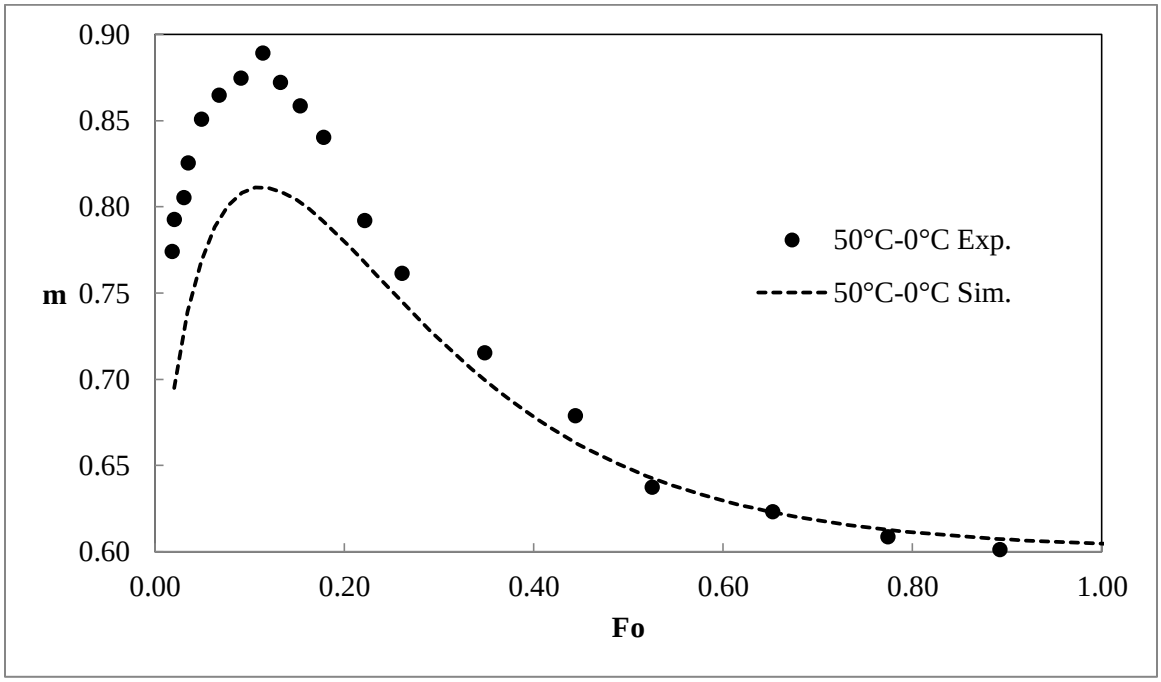

Figure 24: Simulation and Experimental results of Fractional Coverage vs. Fourier Number for OD (0.9525 cm) and T_i-T_B (50°C-0°C).

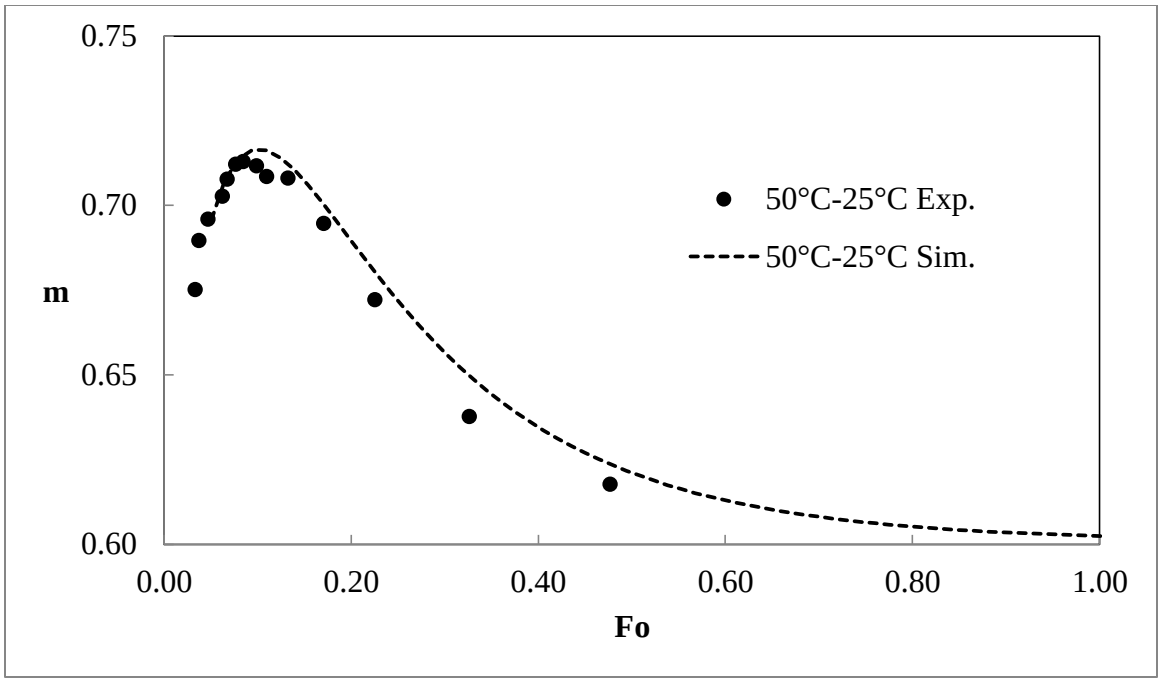

Figure 25: Simulation and Experimental results of Fractional Coverage vs. Fourier Number for OD (0.635 cm) and T_i-T_B (50°C-25°C).

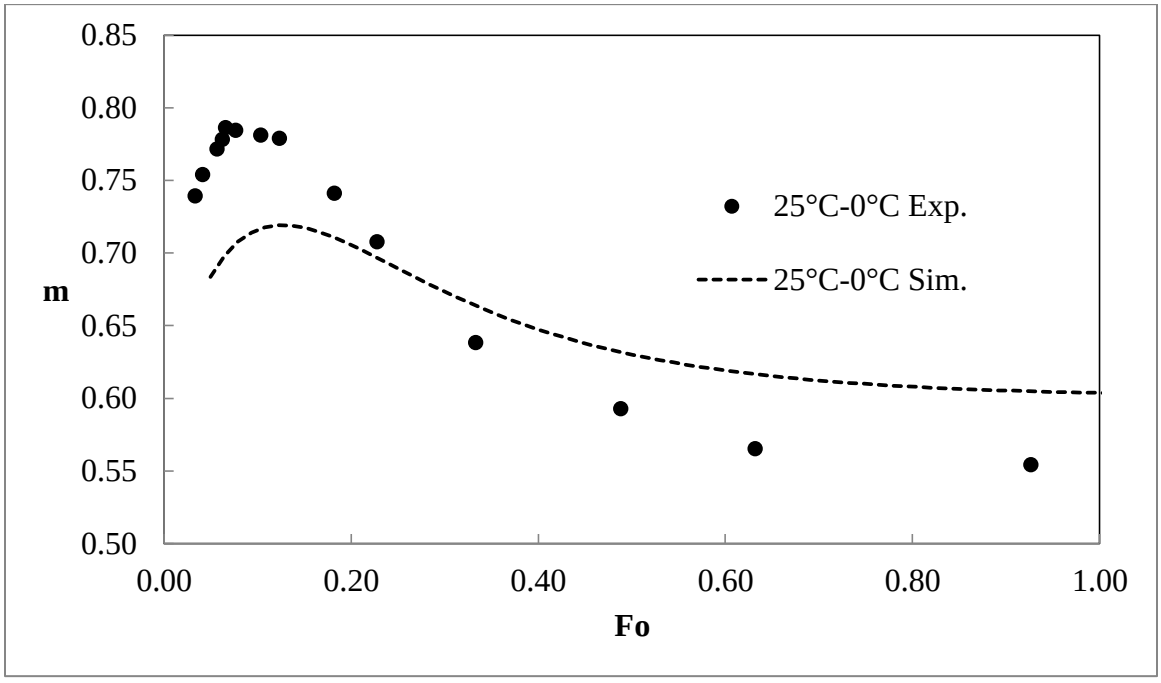

Figure 26: Simulation and Experimental results of Fractional Coverage vs. Fourier Number for OD (0.635 cm) and T_i-T_B (25°C-0°C).

Code 1: Newtonian Model for PBH-300

```
%************************************************************************
%
%   PROJECT:          GAS ASSISTED INJECTION MOLDING
%   PROGRAM NAME:     HTSM 13-1
%   PURPOSE:          NEW ITERATION METHOD USING PBH-300
%   DATE:             01/14/2016
%
%************************************************************************
clear all
clc
%% Initial Conditions

% Number of nodes
n = 21;

% time measurements
%t = [5 10 20 30 40 50 60 90 120 180 240 480]; %sec
t = linspace(5,500,250); %sec

% time increments
dt = 0.0001;

% Temperature of cooling fluid
Tb = 25; %degC

% Temperature of polymer at center of tube
Ti = 65; %degC

%Fractional coverage asymptotic value
m_n = 0.6;

%% Physical Properties of the tube

% External Diameter
OD = 1/2 * 0.0254; %m
%OD = 3/8 * 0.0254; %m
%OD = 1/4 * 0.0254; %m

% Internal Diameter
ID = 1.0922 * 10^-2; %m
%ID = 0.7747 * 10^-2; %m
%ID = 0.4928 * 10^-2; %m

% External Radius
R02 = OD * 0.5; %m

% Internal Radius
R01 = ID * 0.5; %m

% Length of tube
L = 0.305; %m
%L = 0.22; %m

% Gravitational Acceleration
```

```
g = 9.81; %m/s^2

% Diameter
D = OD; %m

% Thermal Conductivity
k_s = 16; %W(m*K)

%Density
den_s = 7850; %kg/m^3

%Specific Heat
Cp_s = 514.5; %J/(kg*K)

% Thermal Diffusivity
a_s = k_s / (den_s * Cp_s); %m^2/s

% Step change in tube
dr2 = (R02 - R01) / (n - 1);

%% Physical Properties of the polymer

% Specific Heat
Cp_p = 2100; %J/(kg*K)

% Thermal Conductivity
k_p = 0.1125; %W/(m*K)

% Density
den_p = -0.58 * Ti + 903; %kg/m^3

% Thermal Diffusivity
a_p = k_p / (den_p * Cp_p); %m^2/s

% Flow Activation Energy
delta_H_R = 8575; %K

% Frequency factor
A = 72.2 / exp(delta_H_R / (25+273.13)); %kg/(m*s)

% Pressure gradient
delta_p = 1.013 * 10^6; %Pa

% Step change in polymer
dr1 = R01 / (n - 1);

n_dr1 = 1.0 / (n - 1);

%% Physical Properties of the cooling fluid

% Specific Heat
Cp_f = 10^-6 * Tb^4 - 0.0004 * Tb^3 + 0.0488 * Tb^2 - 2.2527 * Tb +
4212; %J/(kg*K)

% Viscosity
u_f = (4 * 10^-8 * Tb^4 - 10^-5 * Tb^3 + 0.001 * Tb^2 - 0.0557 * Tb + 1.7832)
* 10^-3; %kg/(m*s)
```

```
% Density
den_f = -0.0036 * Tb^2 - 0.0656 * Tb + 1000.4; %kg/m^3

% Thermal Conductivity
k_f = -9 * 10^-6 * Tb^2 + 0.0021 * Tb + 0.5607; %W/(m*K)

% Coefficient of Volume Expansion
b = (0.0036 * 2 * Tb + 0.0656) / den_f; %1/K

% Thermal Diffusivity
a_f = k_f / (den_f * Cp_f); %m^2/s

%% Heat Transfer outside the tube wall

% Initializing temperature profiles
Temp_p = zeros(n,2);
Temp_s = zeros(n,2);
Temp_fp = zeros(n,length(t));
Temp_fs = zeros(n,length(t));

% Initializing radial position profiles
for i = n : -1 : 1
    r_s(i) = R01 + dr2 * (i-1);
end
for i = n : -1 : 1
    r_p(i) = dr1 * (i-1);
end

% Boundary Condition 1: @ t=0, T_f(r,0) = Ti
Temp_p = Temp_p + Ti;

% Boundary Condition 2: @ t=0, T_s(r,0) = Ti
Temp_s = Temp_s + Ti;

% Calculation at R01
i=n;
j=1;

% Grashof Number
Gr = D^3 * den_f^2 * g * b * (Temp_s(i,j)-Tb) / u_f^2;

% Prandtl Number
Pr = Cp_f * u_f / k_f;

% Raleigh Number
Ra = Gr * Pr;

% Nusselt Number
Nu = (0.6 + 0.387*Ra^(1/6)/(1 + (0.559/Pr)^(9/16))^(8/27))^2;

% Heat Transfer Coefficient
h = k_f * Nu / D;

%Initialization of counter variables
count = 0;
```

```
for k = 1 : length(t)

    disp(length(t)-k+1)

    while count <= t(k)

        %counter variables incrementation
        count = count + dt;

        % Boundary Condition 3: @ r=R02, -ks*dT/dr = h(Ts(R02,t)-Tb)
        i=n;
        j=1;
        Temp_s(i,j+1) = Temp_s(i,j) + (2*a_s*dt/dr2^2)*(Temp_s(i-1,j)-
Temp_s(i,j)-((R02+(dr2/2))/R02)*(h/k_s)*dr2*(Temp_s(i,j)-Tb));

        % Temperature profile in Solid
        for i = n-1 : -1 : 2

            Temp_s(i,j+1) = Temp_s(i,j) +
(a_s*dt/(r_s(i)*dr2^2))*(((r_s(i)+r_s(i-1))/2)*Temp_s(i-1,j)-
2*r_s(i)*Temp_s(i,j)+((r_s(i+1)+r_s(i))/2)*Temp_s(i+1,j));

        end

        % Boundary Condition 4: @ r=R01, solid-liquid interface energy
balance
        i = 1;
        Temp_s(i,j+1) = Temp_s(i,j) + (2*k_s*dt*(R01+dr2/2)*(Temp_s(i+1,j)-
Temp_s(i,j))/(dr2*((R01+dr2/4)*den_s*Cp_s*dr2+(R01-dr1/4)*(-0.58 * Temp_p(n,j)
+ 903)*Cp_p*dr1)))+(2*k_p*dt*(R01-dr1/2)*(Temp_p(n-i,j)-
Temp_s(i,j))/(dr1*((R01+dr2/4)*den_s*Cp_s*dr2+(R01-dr1/4)*(-0.58 * Temp_p(n,j)
+ 903)*Cp_p*dr1)));

        Temp_p(n,j+1) = Temp_s(i,j+1);

        % Temperature profile in Polymer
        for i = n-1 : -1 : 2

            Temp_p(i,j+1) = Temp_p(i,j) + ((k_p/((-0.58 * Temp_p(i,j) +
903)*Cp_p))*dt/(r_p(i)*dr1^2))*(((r_p(i)+r_p(i-1))/2)*Temp_p(i-1,j)-
2*r_p(i)*Temp_p(i,j)+((r_p(i+1)+r_p(i))/2)*Temp_p(i+1,j));

        end

        % Boundary Condition 5: @r=0, kdT/dr = 0
        i=1;
        Temp_p(i,j+1) = Temp_p(i,j) + (4*(k_p/((-0.58 * Temp_p(i,j) +
903)*Cp_p))*dt/dr1^2)*(Temp_p(i+1,j)-Temp_p(i,j));

        for i = 1 : n

            Temp_s(i,j) = Temp_s(i,j+1);
            Temp_p(i,j) = Temp_p(i,j+1);

        end

    end
```

```
    % Creating Temperature Profiles for Polymer and Solid
    for i = 1 : n

        Temp_fp(i,k) = Temp_p(i,j);
        Temp_fs(i,k) = Temp_s(i,j);

    end
end

%% Temperature Calculation using Multi-temperature profiles

position_n = 9;

position_r(1)= 0.77748*R01;
position_r(2)=0.775875115*R01;
position_r(3)=0.757790661*R01;
position_r(4)=0.721989134*R01;
position_r(5)=0.665617809*R01;
position_r(6)=0.58273747*R01;
position_r(7)=0.45921235*R01;
position_r(8)=0.333237929*R01;
position_r(9)=0.238591995*R01;

for i =1 : n
    mark(i) = 1;
end

radius(n) = R01;

for k = 1 : length(t)

    for i = 1 : position_n

        squeeze_ratio = (R01 - position_r(i)) / R01;

        for j = 1 : n

            r_local(j) = position_r(i) + (j - 1) * dr1 * squeeze_ratio;

        end

        for q = 1 : n

            for p = 1 : n-1

                if (r_local(p) < radius(q))

                    if (r_local(p+1) > radius(q))

                        Temp_in = Temp_fp(p+1,k) + (r(q) - r_local(p+1)) /
(r_local(p) - r_local(p+1)) * (Temp_fp(p,k) - Temp_fp(p+1,k));

                        Temp_fp(q,k) = Temp_fp(q,k) + Temp_in;

                        mark(q) = mark(q) + 1;
```

```
                    end

                end

            end

        end

    end

    for i = 1 : n

        Temp_fp(i,k) = Temp_fp(i,k) / mark(i);

        mark(i) = 1;

    end

end

%% Velocity Profile Calculation

%initialize velocity matrix
vel = zeros(n,length(t));

%velocity profile calculation
for k = 1 : length(t)

    %equation 4.33
    vel(n,k) = 0;

    for i = n-1 : -1 : 1

        f(i) =
(n_dr1/2)*delta_p/(2*L*A)*((r_p(i+1)/exp(delta_H_R/(Temp_fp(i+1,k)+273.13)))+
(r_p(i)/exp(delta_H_R/(Temp_fp(i,k)+273.13))));
        vel(i,k) = vel(i+1,k) + f(i);

    end

end

%normalizing the velocity profile
n_vel = zeros(n,length(t));

for k = 1 : length(t)

    for i = 1 : n

        n_vel(i,k) = vel(i,k) / vel(1,k);

    end

end

%normalizing the radial position
n_radius = zeros(2*n,1);
```

```
for i = 1 : n
    n_radius(i) = r_p(i) / R01;
end

for i = 1 : n
    n_radius(i+n) = r_s(i) / R01;
end

%% Fractional Coverage Calculation

%initialize vectors
m = zeros(length(t),1);
Rx = zeros(length(t),1);

for k = 1 : length(t)

    %initializing summation variables
    sum = 0;

    for i = 1 : n-1

        %integrate through u*
        %sum = sum + (n_dr1 / 2) * (n_vel(i,k) + n_vel(i+1,k));
        %integrate through ru*
        sum = sum + (n_dr1 / 2) * (n_radius(i)*n_vel(i,k) +
n_radius(i+1)*n_vel(i+1,k));

    end

    %integrate through u*
    %Rx(k) = R01 * (1.5 * sum);
    %integrate through ru*
    Rx(k) = R01 * sqrt(4 * sum);

    %Fractional coverage when integrating through ru*
    m(k) = 1 - (1 - m_n) * (Rx(k) / R01)^2;

end

%% Experimental Data

if OD / 0.0254 == 1/2;
%**************************************************************************
%   Temperature Gradient:   25C - 0C
%   Tube Size:              1/2"
%**************************************************************************

if Tb == 0 && Ti == 25

%Experimental Average Delay Time
t_exp = [5.65 7.245 8.545 12.265 17.075 22.395 27.975 33.3 38.35 42.965 47.6
52.79 59.81 78.845 95.17 125.35 186.6 247.445 307.515 366.99 426.17 486];

%Experimental Fractional Convergence
m_exp = [0.681675205 0.690412649 0.727022936 0.74366757 0.756240001
0.773033315 0.781406575 0.785302046 0.785784681 0.781528051 0.777379378
```

```
0.769799713 0.756578224 0.744186475 0.713496516 0.682425799 0.664492387
0.638490875 0.624570173 0.620707844 0.61644331 0.606841874];

fname='m(25-0,0.5).dat';

%************************************************************************
%   Temperature Gradient:   25C - 50C
%   Tube Size:              1/2"
%************************************************************************

elseif Tb == 50 && Ti == 25

%Experimental Average Delay Time
t_exp = [5.455 7.555 9.615 12.01 22.01 31.985 36.975 41.905 46.825 61.865
91.695 121.535 181.42 241.485 362.06 482.145];

%Experimental Fractional Convergence
m_exp = [0.471142568 0.456079274 0.429313263 0.424015411 0.414329041
0.405297062 0.410117081 0.416558518 0.42485459 0.442387187 0.472673271
0.506779001 0.536943734 0.561096841 0.575035709 0.579147292];

fname='m(25-50,0.5).dat';

%************************************************************************
%   Temperature Gradient:   35C - 25C
%   Tube Size:              1/2"
%************************************************************************

elseif Tb == 25 && Ti == 35

%Experimental Average Delay Time
t_exp = [6.71 7.73 12.48 22.065 32.235 37.105 42.07 48.25 61.85 91.875 122.1
182.265 242.265 362.175 482.25];

%Experimental Fractional Convergence
m_exp = [0.624713941 0.627619048 0.63819143 0.649094815 0.657936928
0.659108655 0.663238354 0.65908018 0.65366063 0.648955016 0.641569003
0.623405244 0.617466562 0.61636888 0.612840328];

fname='m(35-25,0.5).dat';

%************************************************************************
%   Temperature Gradient:   50C - 0C
%   Tube Size:              1/2"
%************************************************************************

elseif Tb == 0 && Ti == 50

%Experimental Average Delay Time
t_exp = [4.15 5.13 7.41 9.43 11 14.32 15.1 18 20.07 23.27 27.23 32.27 34.16
36.23 42.03 46.19 51.51 60 75.21 90.46 105.23 120.15 140.2 160.11 180 200.22
220.24 240 270.29 300.18 330.27 360.27 390.13];

%Experimental Fractional Convergence
m_exp = [0.6957 0.716 0.7249 0.7452 0.765 0.7849 0.7917 0.8067 0.8119 0.8233
0.8274 0.8347 0.8387 0.8399 0.8415 0.8381 0.8385 0.8285 0.7847 0.7683 0.757
```

```
0.739 0.7224 0.6916 0.6842 0.6632 0.6531 0.6337 0.632 0.6239 0.6234 0.6121
0.61];

fname='m(50-0,0.5).dat';

%************************************************************************
%   Temperature Gradient:   50C - 25C
%   Tube Size:              1/2"
%************************************************************************

elseif Tb == 25 && Ti == 50

%Experimental Average Delay Time
t_exp = [5.795 7.105 12.055 22.63 34.85 37.01 42.9 47.31 63 93 123.16 183.75
244 363.5 483.175];

%Experimental Fractional Convergence
m_exp = [0.656450375 0.663764856 0.686230764 0.714323196 0.736689314
0.737291868 0.743132008 0.737513112 0.731326993 0.709901593 0.681940572
0.650073741 0.634634271 0.614689833 0.613642713];

fname='m(50-25,0.5).dat';

%************************************************************************
%   Temperature Gradient:   65C - 25C
%   Tube Size:              1/2"
%************************************************************************

elseif Tb == 25 && Ti == 65

%Experimental Average Delay Time
t_exp = [4.66 6.075 11.17 21.71 31.735 36.95 41.935 47.395 62.76 77.755 93.06
122.945 182.94 243 362.395 482.315 602.345];

%Experimental Fractional Convergence
m_exp = [0.65503801 0.684064315 0.726081414 0.758627351 0.769766975
0.765677069 0.762767524 0.754104633 0.730933238 0.724348155 0.705838631
0.691798488 0.67274862 0.660508981 0.641360296 0.629218741 0.623456595];

fname='m(65-25,0.5).dat';

end

elseif OD == 3/8*0.0254 && Tb == 0 && Ti == 50

%************************************************************************
%   Temperature Gradient:   50C - 0C
%   Tube Size:              3/8"
%************************************************************************

%Experimental Average Delay Time
t_exp = [4.57317 5.0813 7.62195 8.63821 12.19512 16.76829 22.35772 27.94715
32.52033 37.60163 43.69919 54.36992 64.02439 85.36585 108.78806 128.63689
159.7579 189.65367 218.56925 263.65796 480.27977];

%Experimental Fractional Convergence
```

```
m_exp = [0.77388 0.79224 0.8051 0.82531 0.85041 0.86449 0.87429 0.88898
0.87184 0.85837 0.84 0.79163 0.76102 0.7151 0.67853 0.6372 0.62302 0.60826
0.60088 0.59852 0.59999];

fname='m(50-0,0.375).dat';

elseif OD / 0.0254 == 1/4;

%************************************************************************
%   Temperature Gradient:   25C - 0C
%   Tube Size:              1/4"
%************************************************************************

if Tb == 0 && Ti == 25

%Experimental Average Delay Time
t_exp = [3.38341 4.2064 5.76094 6.3096 6.67538 7.7727 10.42456 12.43632
18.38014 22.95232 33.55976 49.1966 63.64468 93.36381];

%Experimental Fractional Convergence
m_exp = [0.73918 0.75353 0.77127 0.77779 0.78614 0.78431 0.78091 0.77882
0.74095 0.70753 0.63808 0.59235 0.5649 0.55383];

fname='m(25-0,0.25).dat';

%************************************************************************
%   Temperature Gradient:   50C - 25C
%   Tube Size:              1/4"
%************************************************************************

elseif Tb == 25 && Ti == 50

%Experimental Average Delay Time
t_exp = [3.30231 3.71609 4.70917 6.19879 6.69532 7.60564 8.43321 9.84007
10.9159 13.15032 16.95712 22.33628 32.34981 47.24596 62.80417 92.09993
122.38878];

%Experimental Fractional Convergence
m_exp = [0.67508 0.68945 0.69586 0.70253 0.70766 0.71202 0.71279 0.71151
0.70843 0.70791 0.69458 0.672 0.63763 0.61762 0.58325 0.57427 0.57093];

fname='m(50-25,0.25).dat';

end

end

%% Plotting Fractional Coverage vs Time

figure(1)

plot(t, m,'k', t_exp, m_exp,'rs')

xlabel('Delay Time (sec)')

ylabel('Fractional Coverage - m')
```

```
legend('New Model','Experimental data')

%% Fourier Number

Fo = a_p*t/R01^2;

Fo_exp = a_p*t_exp/R01^2;

figure(2)

plot(Fo, m,'k', Fo_exp, m_exp,'rs')

xlabel('Fourier Number - Fo')

ylabel('Fractional Coverage - m')

legend('New Model','Experimental data')

%% Biot Number

Bi = h*R01/k_p;

%*************************************************************************
%
%                                 CODE END
%
%*************************************************************************
```

Code 2: Newtonian Model for DC-200

```
%************************************************************************
%
%   PROJECT:          GAS ASSISTED INJECTION MOLDING
%   PROGRAM NAME:     HTSM 15-0
%   PURPOSE:          NEW ITERATION METHOD USING DC-200
%   DATE:             01/14/2016
%
%************************************************************************
clear all
clc
%% Initial Conditions

% Number of nodes
n = 21;

% time measurements
%t = [5 10 20 30 40 50 60 90 120 180 240 480]; %sec
t = linspace(5,500,100); %sec

% time increments
dt = 0.0001;

% Temperature of cooling fluid
Tb = 25; %degC

% Temperature of polymer at center of tube
Ti = 65; %degC

%Fractional coverage asymptotic value
m_n = 0.6;

%% Physical Properties of the tube

% External Diameter
OD = 1/2 * 0.0254; %m
%OD = 3/8 * 0.0254; %m
%OD = 1/4 * 0.0254; %m

% Internal Diameter
ID = 1.0922 * 10^-2; %m
%ID = 0.7747 * 10^-2; %m
%ID = 0.4928 * 10^-2; %m

% External Radius
R02 = OD * 0.5; %m

% Internal Radius
R01 = ID * 0.5; %m

% Length of tube
L = 0.305; %m
%L = 0.22; %m

% Gravitational Acceleration
```

```
g = 9.81; %m/s^2

% Diameter
D = OD; %m

% Thermal Conductivity
k_s = 16; %W(m*K)

%Density
den_s = 7850; %kg/m^3

%Specific Heat
Cp_s = 514.5; %J/(kg*K)

% Thermal Diffusivity
a_s = k_s / (den_s * Cp_s); %m^2/s

% Step change in tube
dr2 = (R02 - R01) / (n - 1);

%% Physical Properties of the polymer

% Specific Heat
Cp_p = 1456; %J/(kg*K)

% Thermal Conductivity
k_p = 0.1547; %W/(m*K)

% Density
den_p = 971; %kg/m^3
%den_p = -0.58 * Ti + 903; %kg/m^3

% Thermal Diffusivity
a_p = k_p / (den_p * Cp_p); %m^2/s

% Flow Activation Energy
delta_H_R = 1090; %K

% Frequency factor
A = 20.5 / exp(delta_H_R / (25+273.13)); %kg/(m*s)

% Pressure gradient
delta_p = 1.013 * 10^6; %Pa

% Step change in polymer
dr1 = R01 / (n - 1);

n_dr1 = 1.0 / (n - 1);

%% Physical Properties of the cooling fluid

% Specific Heat
Cp_f = 10^-6 * Tb^4 - 0.0004 * Tb^3 + 0.0488 * Tb^2 - 2.2527 * Tb +
4212; %J/(kg*K)

% Viscosity
```

```
u_f = (4 * 10^-8 * Tb^4 - 10^-5 * Tb^3 + 0.001 * Tb^2 - 0.0557 * Tb + 1.7832)
* 10^-3; %kg/(m*s)

% Density
den_f = -0.0036 * Tb^2 - 0.0656 * Tb + 1000.4; %kg/m^3

% Thermal Conductivity
k_f = -9 * 10^-6 * Tb^2 + 0.0021 * Tb + 0.5607; %W/(m*K)

% Coefficient of Volume Expansion
b = (0.0036 * 2 * Tb + 0.0656) / den_f; %1/K

% Thermal Diffusivity
a_f = k_f / (den_f * Cp_f); %m^2/s

%% Heat Transfer outside the tube wall

% Initializing temperature profiles
Temp_p = zeros(n,2);
Temp_s = zeros(n,2);
Temp_fp = zeros(n,length(t));
Temp_fs = zeros(n,length(t));

% Initializing radial position profiles
for i = n : -1 : 1
    r_s(i) = R01 + dr2 * (i-1);
end
for i = n : -1 : 1
    r_p(i) = dr1 * (i-1);
end

% Boundary Condition 1: @ t=0, T_f(r,0) = Ti
Temp_p = Temp_p + Ti;

% Boundary Condition 2: @ t=0, T_s(r,0) = Ti
Temp_s = Temp_s + Ti;

% Calculation at R01
i=n;
j=1;

% Grashof Number
Gr = D^3 * den_f^2 * g * b * (Temp_s(i,j)-Tb) / u_f^2;

% Prandtl Number
Pr = Cp_f * u_f / k_f;

% Raleigh Number
Ra = Gr * Pr;

% Nusselt Number
Nu = (0.6 + 0.387*Ra^(1/6)/(1 + (0.559/Pr)^(9/16))^(8/27))^2;

% Heat Transfer Coefficient
h = k_f * Nu / D;

%Initialization of counter variables
```

```
count = 0;

for k = 1 : length(t)

    disp(length(t)-k+1)

    while count <= t(k)

        %counter variables incrementation
        count = count + dt;

        % Boundary Condition 3: @ r=R02, -ks*dT/dr = h(Ts(R02,t)-Tb)
        i=n;
        j=1;
        Temp_s(i,j+1) = Temp_s(i,j) + (2*a_s*dt/dr2^2)*(Temp_s(i-1,j)-
Temp_s(i,j)-((R02+(dr2/2))/R02)*(h/k_s)*dr2*(Temp_s(i,j)-Tb));

        % Temperature profile in Solid
        for i = n-1 : -1 : 2

            Temp_s(i,j+1) = Temp_s(i,j) +
(a_s*dt/(r_s(i)*dr2^2))*(((r_s(i)+r_s(i-1))/2)*Temp_s(i-1,j)-
2*r_s(i)*Temp_s(i,j)+((r_s(i+1)+r_s(i))/2)*Temp_s(i+1,j));

        end

        % Boundary Condition 4: @ r=R01, solid-liquid interface energy
balance
        i = 1;
        Temp_s(i,j+1) = Temp_s(i,j) + (2*k_s*dt*(R01+dr2/2)*(Temp_s(i+1,j)-
Temp_s(i,j))/(dr2*((R01+dr2/4)*den_s*Cp_s*dr2+(R01-dr1/4)*(-0.58 * Temp_p(n,j)
+ 903)*Cp_p*dr1)))+(2*k_p*dt*(R01-dr1/2)*(Temp_p(n-i,j)-
Temp_s(i,j))/(dr1*((R01+dr2/4)*den_s*Cp_s*dr2+(R01-dr1/4)*(-0.58 * Temp_p(n,j)
+ 903)*Cp_p*dr1)));

        Temp_p(n,j+1) = Temp_s(i,j+1);

        % Temperature profile in Polymer
        for i = n-1 : -1 : 2

            Temp_p(i,j+1) = Temp_p(i,j) + ((k_p/((-0.58 * Temp_p(i,j) +
903)*Cp_p))*dt/(r_p(i)*dr1^2))*(((r_p(i)+r_p(i-1))/2)*Temp_p(i-1,j)-
2*r_p(i)*Temp_p(i,j)+((r_p(i+1)+r_p(i))/2)*Temp_p(i+1,j));

        end

        % Boundary Condition 5: @r=0, kdT/dr = 0
        i=1;
        Temp_p(i,j+1) = Temp_p(i,j) + (4*(k_p/((-0.58 * Temp_p(i,j) +
903)*Cp_p))*dt/dr1^2)*(Temp_p(i+1,j)-Temp_p(i,j));

        for i = 1 : n

            Temp_s(i,j) = Temp_s(i,j+1);
            Temp_p(i,j) = Temp_p(i,j+1);

        end
```

```
    end

    % Creating Temperature Profiles for Polymer and Solid
    for i = 1 : n

        Temp_fp(i,k) = Temp_p(i,j);
        Temp_fs(i,k) = Temp_s(i,j);

    end
end

%% Temperature Calculation using Multi-temperature profiles

position_n = 9;

position_r(1)= 0.77748*R01;
position_r(2)=0.775875115*R01;
position_r(3)=0.757790661*R01;
position_r(4)=0.721989134*R01;
position_r(5)=0.665617809*R01;
position_r(6)=0.58273747*R01;
position_r(7)=0.45921235*R01;
position_r(8)=0.333237929*R01;
position_r(9)=0.238591995*R01;

for i =1 : n
    mark(i) = 1;
end

radius(n) = R01;

for k = 1 : length(t)

    for i = 1 : position_n

        squeeze_ratio = (R01 - position_r(i)) / R01;

        for j = 1 : n

            r_local(j) = position_r(i) + (j - 1) * dr1 * squeeze_ratio;

        end

        for q = 1 : n

            for p = 1 : n-1

                if (r_local(p) < radius(q))

                    if (r_local(p+1) > radius(q))

                        Temp_in = Temp_fp(p+1,k) + (r(q) - r_local(p+1)) /
(r_local(p) - r_local(p+1)) * (Temp_fp(p,k) - Temp_fp(p+1,k));

                        Temp_fp(q,k) = Temp_fp(q,k) + Temp_in;
```

```
                        mark(q) = mark(q) + 1;

                    end

                end

            end

        end

    end

    for i = 1 : n

        Temp_fp(i,k) = Temp_fp(i,k) / mark(i);

        mark(i) = 1;

    end

end

%% Velocity Profile Calculation

%initialize velocity matrix
vel = zeros(n,length(t));

%velocity profile calculation
for k = 1 : length(t)

    %equation 4.33
    vel(n,k) = 0;
    vel1(n,k) = 0;

    for i = n-1 : -1 : 1

        f(i) =
(n_dr1/2)*delta_p/(2*L*A)*((r_p(i+1)/exp(delta_H_R/(Temp_fp(i+1,k)+273.13)))+
(r_p(i)/exp(delta_H_R/(Temp_fp(i,k)+273.13))));
        vel(i,k) = vel(i+1,k) + f(i);

    end

end

%normalizing the velocity profile
n_vel = zeros(n,length(t));

for k = 1 : length(t)

    for i = 1 : n
        %0.95*
        n_vel(i,k) = vel(i,k) / vel(1,k);

    end

end
```

```
%normalizing the radial position
n_radius = zeros(2*n,1);

for i = 1 : n
    n_radius(i) = r_p(i) / R01;
end

for i = 1 : n
    n_radius(i+n) = r_s(i) / R01;
end

%% Fractional Coverage Calculation

%initialize vectors
m = zeros(length(t),1);
Rx = zeros(length(t),1);

for k = 1 : length(t)

    %initializing summation variables
    sum = 0;

    for i = 1 : n-1

        %integrate through u*
        %sum = sum + (n_dr1 / 2) * (n_vel(i,k) + n_vel(i+1,k));
        %integrate through ru*
        sum = sum + (n_dr1 / 2) * (n_radius(i)*n_vel(i,k) +
n_radius(i+1)*n_vel(i+1,k));

    end

    %integrate through u*
    %Rx(k) = R01 * (1.5 * sum);
    %integrate through ru*
    Rx(k) = R01 * sqrt(4 * sum);

    %Fractional coverage when integrating through ru*
    m(k) = 1 - (1 - m_n) * (Rx(k) / R01)^2;

end

%% Experimental Data

%************************************************************************
%   Temperature Gradient:   65C - 25C
%   Tube Size:              1/2"
%************************************************************************

%Experimental Average Delay Time Corrected
t_exp = [6.675 11.7 22.875 31.21 41.39 51.51 61.53 91.545 121.5 241.56 481.5];

%Experimental Fractional Convergence Corrected
m_exp = [0.61399464 0.619600588 0.627505378 0.628231328 0.629521906
0.627263395 0.621092819 0.613228359 0.601814811 0.600080597 0.594797293];
```

```
%% Plotting Fractional Coverage vs Time

figure(1)

plot(t, m,'k', t_exp, m_exp,'rs')

xlabel('Delay Time (sec)')

ylabel('Fractional Coverage - m')

legend('New Model','Experimental data')

%% Fourier Number

Fo = a_p*t/R01^2;

Fo_exp = a_p*t_exp/R01^2;


figure(2)

plot(Fo, m,'k', Fo_exp, m_exp,'rs')

xlabel('Fourier Number - Fo')

ylabel('Fractional Coverage - m')

legend('New Model','Experimental data')

%% Biot Number

Bi = h*R01/k_p;

%*************************************************************************
%
%                                 CODE END
%
%*************************************************************************
```

Code 3: Non-Newtonian Model

```
%************************************************************************
%
%    PROJECT:          GAS ASSISTED INJECTION MOLDING
%    PROGRAM NAME:     HTSM 18-0
%    PURPOSE:          NEW ITERATION METHOD using Power Law
%    DATE:             01/14/2016
%
%************************************************************************
clear all
clc
%% Initial Conditions

% Number of nodes
nodes = 21;

% time measurements
%t = [5 10 20 30 40 50 60 90 120 180 240 480]; %sec
t = linspace(5,500,100); %sec

% time increments
dt = 0.0001;

% Temperature of cooling fluid
Tb = 25; %degC

% Temperature of polymer at center of tube
Ti = 25; %degC

% power law index
n = 1;

%% Fractional coverage asymptotic value
if n <= 0.3
    m_n = 0.1516*log(n)+0.7259;
elseif n > 0.3
    m_n = -0.1428*n^2+0.2626*n+0.4782;
end

%% Physical Properties of the tube

% External Diameter
OD = 1/2 * 0.0254; %m
%OD = 3/8 * 0.0254; %m
%OD = 1/4 * 0.0254; %m

% Internal Diameter
ID = 1.0922 * 10^-2; %m
%ID = 0.7747 * 10^-2; %m
%ID = 0.4928 * 10^-2; %m

% External Radius
R02 = OD * 0.5; %m

% Internal Radius
```

```
R01 = ID * 0.5; %m

% Length of tube
L = 0.305; %m
%L = 0.22; %m

% Gravitational Acceleration
g = 9.81; %m/s^2

% Diameter
D = OD; %m

% Thermal Conductivity
k_s = 16; %W(m*K)

%Density
den_s = 7850; %kg/m^3

%Specific Heat
Cp_s = 514.5; %J/(kg*K)

% Thermal Diffusivity
a_s = k_s / (den_s * Cp_s); %m^2/s

% Step change in tube
dr2 = (R02 - R01) / (nodes - 1);

%% Physical Properties of the polymer

% Specific Heat
Cp_p = 2100; %J/(kg*K)

% Thermal Conductivity
k_p = 0.1125; %W/(m*K)

% Density
den_p = -0.58 * Ti + 903; %kg/m^3

% Thermal Diffusivity
a_p = k_p / (den_p * Cp_p); %m^2/s

% Flow Activation Energy
delta_H_R = 8575; %K

% Frequency factor
A = 72.2 / exp(delta_H_R / (25+273.13)); %kg/(m*s)

% Pressure gradient
delta_p = 1.013 * 10^6; %Pa

% Step change in polymer
dr1 = R01 / (nodes - 1);

n_dr1 = 1.0 / (nodes - 1);

%% Physical Properties of the cooling fluid
```

```
% Specific Heat
Cp_f = 10^-6 * Tb^4 - 0.0004 * Tb^3 + 0.0488 * Tb^2 - 2.2527 * Tb +
4212; %J/(kg*K)

% Viscosity
u_f = (4 * 10^-8 * Tb^4 - 10^-5 * Tb^3 + 0.001 * Tb^2 - 0.0557 * Tb + 1.7832)
* 10^-3; %kg/(m*s)

% Density
den_f = -0.0036 * Tb^2 - 0.0656 * Tb + 1000.4; %kg/m^3

% Thermal Conductivity
k_f = -9 * 10^-6 * Tb^2 + 0.0021 * Tb + 0.5607; %W/(m*K)

% Coefficient of Volume Expansion
b = (0.0036 * 2 * Tb + 0.0656) / den_f; %1/K

% Thermal Diffusivity
a_f = k_f / (den_f * Cp_f); %m^2/s

%% Heat Transfer outside the tube wall

% Initializing temperature profiles
Temp_p = zeros(nodes,2);
Temp_s = zeros(nodes,2);
Temp_fp = zeros(nodes,length(t));
Temp_fs = zeros(nodes,length(t));

% Initializing radial position profiles
for i = nodes : -1 : 1
    r_s(i) = R01 + dr2 * (i-1);
end
for i = nodes : -1 : 1
    r_p(i) = dr1 * (i-1);
end

% Boundary Condition 1: @ t=0, T_f(r,0) = Ti
Temp_p = Temp_p + Ti;

% Boundary Condition 2: @ t=0, T_s(r,0) = Ti
Temp_s = Temp_s + Ti;

% Calculation at R01
i=nodes;
j=1;

% Grashof Number
Gr = D^3 * den_f^2 * g * b * (Temp_s(i,j)-Tb) / u_f^2;

% Prandtl Number
Pr = Cp_f * u_f / k_f;

% Raleigh Number
Ra = Gr * Pr;

% Nusselt Number
Nu = (0.6 + 0.387*Ra^(1/6)/(1 + (0.559/Pr)^(9/16))^(8/27))^2;
```

```
% Heat Transfer Coefficient
h = k_f * Nu / D;

%Initialization of counter variables
count = 0;

for k = 1 : length(t)

    disp(length(t)-k+1)

    while count <= t(k)

        %counter variables incrementation
        count = count + dt;

        % Boundary Condition 3: @ r=R02, -ks*dT/dr = h(Ts(R02,t)-Tb)
        i=nodes;
        j=1;
        Temp_s(i,j+1) = Temp_s(i,j) + (2*a_s*dt/dr2^2)*(Temp_s(i-1,j)-
Temp_s(i,j)-((R02+(dr2/2))/R02)*(h/k_s)*dr2*(Temp_s(i,j)-Tb));

        % Temperature profile in Solid
        for i = nodes-1 : -1 : 2

            Temp_s(i,j+1) = Temp_s(i,j) +
(a_s*dt/(r_s(i)*dr2^2))*(((r_s(i)+r_s(i-1))/2)*Temp_s(i-1,j)-
2*r_s(i)*Temp_s(i,j)+((r_s(i+1)+r_s(i))/2)*Temp_s(i+1,j));

        end

        % Boundary Condition 4: @ r=R01, solid-liquid interface energy
balance
        i = 1;
        Temp_s(i,j+1) = Temp_s(i,j) + (2*k_s*dt*(R01+dr2/2)*(Temp_s(i+1,j)-
Temp_s(i,j))/(dr2*((R01+dr2/4)*den_s*Cp_s*dr2+(R01-dr1/4)*(-0.58 *
Temp_p(nodes,j) + 903)*Cp_p*dr1)))+(2*k_p*dt*(R01-dr1/2)*(Temp_p(nodes-i,j)-
Temp_s(i,j))/(dr1*((R01+dr2/4)*den_s*Cp_s*dr2+(R01-dr1/4)*(-0.58 *
Temp_p(nodes,j) + 903)*Cp_p*dr1)));

        Temp_p(nodes,j+1) = Temp_s(i,j+1);

        % Temperature profile in Polymer
        for i = nodes-1 : -1 : 2

            Temp_p(i,j+1) = Temp_p(i,j) + ((k_p/((-0.58 * Temp_p(i,j) +
903)*Cp_p))*dt/(r_p(i)*dr1^2))*(((r_p(i)+r_p(i-1))/2)*Temp_p(i-1,j)-
2*r_p(i)*Temp_p(i,j)+((r_p(i+1)+r_p(i))/2)*Temp_p(i+1,j));

        end

        % Boundary Condition 5: @r=0, kdT/dr = 0
        i=1;
        Temp_p(i,j+1) = Temp_p(i,j) + (4*(k_p/((-0.58 * Temp_p(i,j) +
903)*Cp_p))*dt/dr1^2)*(Temp_p(i+1,j)-Temp_p(i,j));

        for i = 1 : nodes
```

```
            Temp_s(i,j) = Temp_s(i,j+1);
            Temp_p(i,j) = Temp_p(i,j+1);

        end

    end

    % Creating Temperature Profiles for Polymer and Solid
    for i = 1 : nodes

        Temp_fp(i,k) = Temp_p(i,j);
        Temp_fs(i,k) = Temp_s(i,j);

    end
end

%% Temperature Calculation using Multi-temperature profiles

position_n = 9;

position_r(1)= 0.77748*R01;
position_r(2)=0.775875115*R01;
position_r(3)=0.757790661*R01;
position_r(4)=0.721989134*R01;
position_r(5)=0.665617809*R01;
position_r(6)=0.58273747*R01;
position_r(7)=0.45921235*R01;
position_r(8)=0.333237929*R01;
position_r(9)=0.238591995*R01;

for i =1 : nodes
    mark(i) = 1;
end

radius(nodes) = R01;

for k = 1 : length(t)

    for i = 1 : position_n

        squeeze_ratio = (R01 - position_r(i)) / R01;

        for j = 1 : nodes

            r_local(j) = position_r(i) + (j - 1) * dr1 * squeeze_ratio;

        end

        for q = 1 : nodes

            for p = 1 : nodes-1

                if (r_local(p) < radius(q))

                    if (r_local(p+1) > radius(q))
```

```
                        Temp_in = Temp_fp(p+1,k) + (r(q) - r_local(p+1)) /
(r_local(p) - r_local(p+1)) * (Temp_fp(p,k) - Temp_fp(p+1,k));

                        Temp_fp(q,k) = Temp_fp(q,k) + Temp_in;

                        mark(q) = mark(q) + 1;

                    end

                end

            end

        end

    end

    for i = 1 : nodes

        Temp_fp(i,k) = Temp_fp(i,k) / mark(i);

        mark(i) = 1;

    end

end

%% Velocity Profile Calculation

%initialize velocity matrix
vel = zeros(nodes,length(t));

%velocity profile calculation
for k = 1 : length(t)

    %equation 4.33
    vel(nodes,k) = 0;

    for i = nodes-1 : -1 : 1

%         f(i) =
(n_dr1/2)*delta_p/(2*L*A)*((r_p(i+1)/exp(delta_H_R/(Temp_fp(i+1,k)+273.13)))+
(r_p(i)/exp(delta_H_R/(Temp_fp(i,k)+273.13))));
%         vel(i,k) = vel(i+1,k) + f(i);
        f(i) =
(n_dr1/2)*(delta_p/(2*L*A))^(1/n)*(((r_p(i+1)/exp(delta_H_R/(Temp_fp(i+1,k)+2
73.13))))^(1/n)+((r_p(i)/exp(delta_H_R/(Temp_fp(i,k)+273.13))))^(1/n));
        vel(i,k) = vel(i+1,k) + f(i);

    end

end

%normalizing the velocity profile
n_vel = zeros(nodes,length(t));

for k = 1 : length(t)
```

```
    for i = 1 : nodes
        %0.95*
        n_vel(i,k) = vel(i,k) / vel(1,k);

    end

end

%normalizing the radial position
n_radius = zeros(2*nodes,1);

for i = 1 : nodes
    n_radius(i) = r_p(i) / R01;
end

for i = 1 : nodes
    n_radius(i+nodes) = r_s(i) / R01;
end

%% Fractional Coverage Calculation

%initialize vectors
m = zeros(length(t),1);
Rx = zeros(length(t),1);

for k = 1 : length(t)

    %initializing summation variables
    sum = 0;

    for i = 1 : nodes-1

        %integrate through u*
        %sum = sum + (n_dr1 / 2) * (n_vel(i,k) + n_vel(i+1,k));
        %integrate through ru*
        sum = sum + (n_dr1 / 2) * (n_radius(i)*n_vel(i,k) +
n_radius(i+1)*n_vel(i+1,k));

    end

    %integrate through u*
    %Rx(k) = R01 * (1.5 * sum);
    %integrate through ru*
    Rx(k) = R01 * sqrt(4 * sum);

    %Fractional coverage when integrating through ru*
    %m(k) = 1 - (1 - m_n) * (Rx(k) / R01)^2;
    m_new = ((m_n-1+((2+(2*n))/(1+(3*n)))))/((2+(2*n))/(1+(3*n)));
    m(k) = 1 - (1 - m_new) * (Rx(k) / R01)^2;

end

%% Fourier Number

Fo = a_p*t/R01^2;
```

```
%% Plot

figure (1)

plot(Fo, m)

xlabel('Fourier Number - Fo')

ylabel('Fractional Coverage - m')

%% Biot Number

Bi = h*R01/k_p;

%*************************************************************************
%
%                                 CODE END
%
%*************************************************************************
```